UNLOCKING THE URBAN PHOTOGRAPHIC RECORD THROUGH 4D SCENE MODELING

A Thesis
Presented to
The Academic Faculty

by

Grant Schindler

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing
College of Computing

Georgia Institute of Technology
August 2010

UNLOCKING THE URBAN PHOTOGRAPHIC RECORD THROUGH 4D SCENE MODELING

Approved by:

Frank Dellaert, Advisor
School of Interactive Computing
College of Computing
Georgia Institute of Technology

Sing Bing Kang
Microsoft Research
Redmond

Irfan Essa
School of Interactive Computing
College of Computing
Georgia Institute of Technology

Steven Seitz
Department of Computer Science and Engineering
University of Washington

Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 30 May 2010

for Joanna

ACKNOWLEDGEMENTS

I owe a debt of gratitude to the many people who have helped me along the way to finishing the work presented here. I must thank my advisor, Frank Dellaert, for conceiving of and giving me the opportunity to work on an amazing project that I truly cared about, for his intense guidance early on, and for the subsequent autonomy of the last few years.

For access to the historical photographs of Atlanta and Manhattan used in this work, I thank the Atlanta History Center and the New York Public Library. Without these images, none of my work would have been possible. I'd also like to thank the many Flickr users who allowed the use of their photographs, and who are credited individually for their photos throughout this dissertation.

My fellow students have enriched my life in many ways, acting as friends, colleagues, and traveling companions. To name but a few: thanks to Ananth Ranganathan for conversations on science fiction, to Michael Kaess for showing me the ropes, to Sang Min Oh for travels in China, to Mingxuan Sun for ray tracing and good humor, to Kai Ni for travels through Japan, to Kevin Quennesson for beautiful animation tools, and to all of the above and many more for being great friends.

Finally, I must thank my family for everything they have done for me. To my parents and my sister, thank you for supporting me in all my endeavors and for starting me down the path that led me to where I am today. And above all, to my wife, Joanna, thank you for your love and support over the years. I couldn't have done it without you.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Vast collections of historical photographs are being digitally archived and placed online, providing an objective record of the last two centuries that remains largely untapped. We propose that time-varying 3D models can pull together and index large collections of images while also serving as a tool of historical discovery, revealing new information about the locations, dates, and contents of historical images. In particular, our goal is to use computer vision techniques to tie together a large set of historical photographs of a given city into a consistent 4D model of the city: a 3D model with time as an additional dimension.

To extract 4D city models from historical images, we must perform inference about the position of cameras and scene structure in both space and time. Traditional structure from motion techniques can be used to deal with the spatial problem, while here we focus on the problem of inferring temporal information: a date for each image and a time interval for which each structural element in the scene persists.

We first formulate this task as a constraint satisfaction problem based on the visibility of structural elements in each image, resulting in a temporal ordering of images. Next, we present methods to incorporate real date information into the temporal inference solution. Finally, we present a general probabilistic framework for estimating all temporal variables in structure from motion problems, including an unknown date for each camera and an unknown time interval for each structural element. Given a collection of images with mostly unknown or uncertain dates, we can use this framework to automatically recover the dates of all images by reasoning probabilistically about the visibility and existence of objects in the scene. We present results for image collections consisting of hundreds of historical images of cities taken over decades of time, including Manhattan and downtown Atlanta.

Chapter I

INTRODUCTION

Vast collections of historical photographs are being digitally archived and placed online, providing an objective record of the last two centuries that remains largely untapped. We propose that time-varying 3D models can pull together and index large collections of images while also serving as a tool of historical discovery, revealing new information about the locations, dates, and contents of historical images. In particular, our goal is to use computer vision techniques to tie together a large set of historical photographs of a given city into a consistent 4D model of the city: a 3D model with time as an additional dimension.

To extract 4D city models from historical images, we must perform inference about the position of cameras and scene structure in both space and time. Traditional structure from motion techniques can be used to deal with the spatial problem, while here we focus on the problem of inferring temporal information: a date for each image and a time interval for which each structural element in the scene persists.

We first formulate this task as a constraint satisfaction problem based on the visibility of structural elements in each image, resulting in a temporal ordering of images. Next, we present methods to incorporate real date information into the temporal inference solution. Finally, we present a general probabilistic framework for estimating all temporal variables in structure from motion problems, including an unknown date for each camera and an unknown time interval for each structural element. Given a collection of images with mostly unknown or uncertain dates, we can use this framework to automatically recover the dates of all images by reasoning probabilistically about the visibility and existence of objects in the scene. We present results for image collections consisting of hundreds of historical images of cities taken over decades of time, including Manhattan and downtown Atlanta.

Figure 1: Images of Atlanta, Georgia, 1864-2010. Large changes in geometry and appearance pose significant challenges to constructing a time-varying 3D model of the city, which we call a 4D city model. In this dissertation, we show that 4D city models can be used to organize the world's historical photographs, and we present methods to build such models from collections of images, including the first computer vision methods for performing temporal inference on reconstructed 3D scenes. *Photos provided by Atlanta History Center, Library of Congress, and Grant Schindler.*

1.1 Thesis Statement

The thesis of this dissertation is as follows:

4D city models serve to both organize and enhance the world's historical photographs by providing spatial and temporal context for every image. Temporal inference algorithms, when applied to reconstructed 3D scenes, enable the automatic construction of 4D city models directly from images.

In the remainder of this dissertation, we will support this thesis statement by (1) developing a formal representation of time in structure from motion problems, (2) presenting three algorithms for solving temporal inference, (3) detailing a pipeline for automatically building 4D city models, (4) introducing a method of interacting with 4D models, and (5) demonstrating 4D models of Atlanta, Manhattan, and Seoul.

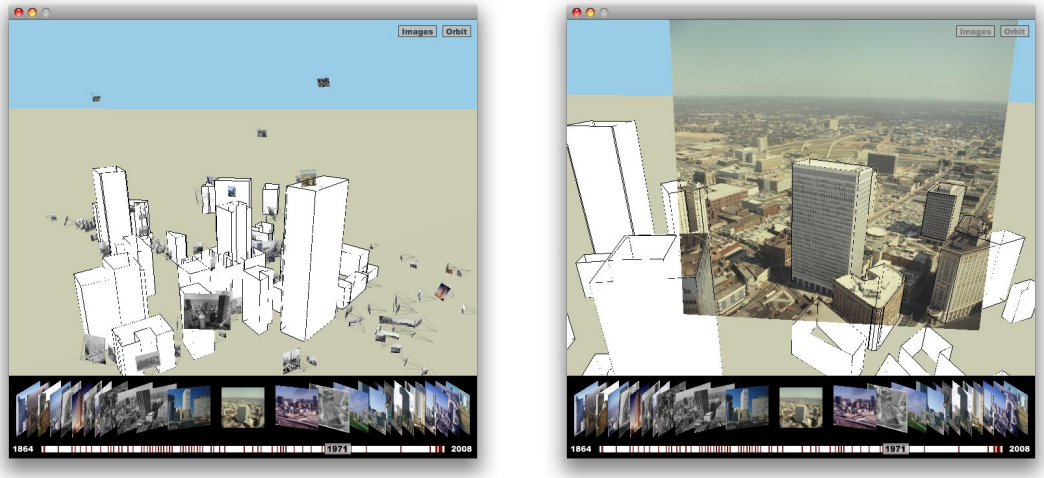


Figure 2: 4D City Model. A bird's eye view of Atlanta in 1971 (left) and the same model from the viewpoint of a selected 1971 photograph (right). We show that 4D city models serve as an effective means of organizing historical photographs and providing context, both spatially and temporally.

1.2 Motivation

There is a need to organize the growing number of historical and modern photographs being digitized and put online. We propose that 4D models – time-varying 3D models of cities – can serve a number of important functions. 4D models can serve to:

- Organize photo collections
- Contextualize individual photographs
- Visualize the past
- Uncover historical details

Furthermore, placing the world's historical photographs into 4D models will contribute to the goals of preserving, understanding, and appreciating the past. In the long term, we believe every urban historical photograph on record will be placed into a 4D model of the type described in this dissertation.

Historical photographs are currently distributed across a wide variety of internet locations. In the case of Atlanta, such photos (see Figure 1) can be found through the Atlanta

History Center, the Library of Congress, the Digital Library of Georgia, and Flickr, just to name a few. Currently, these collections are completely separate entities, with no means of finding similar images between collections, images of the same buildings at different times, or images taken from the same location at the same time, for example. The collections do not know about each other, and it is difficult to form a solid understanding of a given city in a given decade without spending a significant amount of time in each collection. By registering images to a 4D model (Figure 2), the barriers between image collections fall and one can easily get a birds-eye view of all the images from a given era, or transition between two views of the same building that come from different sources.

When viewing a photograph of the past, it can be difficult to get a sense for where the image was *really* taken. In some cases, the entire city has changed beyond recognition. In other cases, it can be difficult to decide if the photo is looking north or south on a given street. By registering the image to a 4D model, the context of the photograph becomes clear, both in space and time. Buildings that were not even in the original photograph become visible around the edges of the image. The viewer can look down to discover that the photographer was standing on a rooftop not visible in the image. In the case of old photographs sharing no common scenery with the present, one can even make visible the 3D models of modern buildings to get a sense of where the photo is positioned with respect to modern day structures.

We also propose that 4D city models can serve as a tool of historical discovery. By ensuring that every photo is registered to the same model in a mathematically consistent way, we can reveal information about the precise locations, dates, and contents of photographs that would have been unrecoverable without such a 4D model. As an example, we have been able to determine the precise latitude and longitude of a set of 1864 images captured by Civil War photographer George Barnard, in spite of the fact that no structures (either natural or man-made) pictured in those 1864 photos still exist today. However, because at each stage of history, there has been overlap between the structures of one decade and the

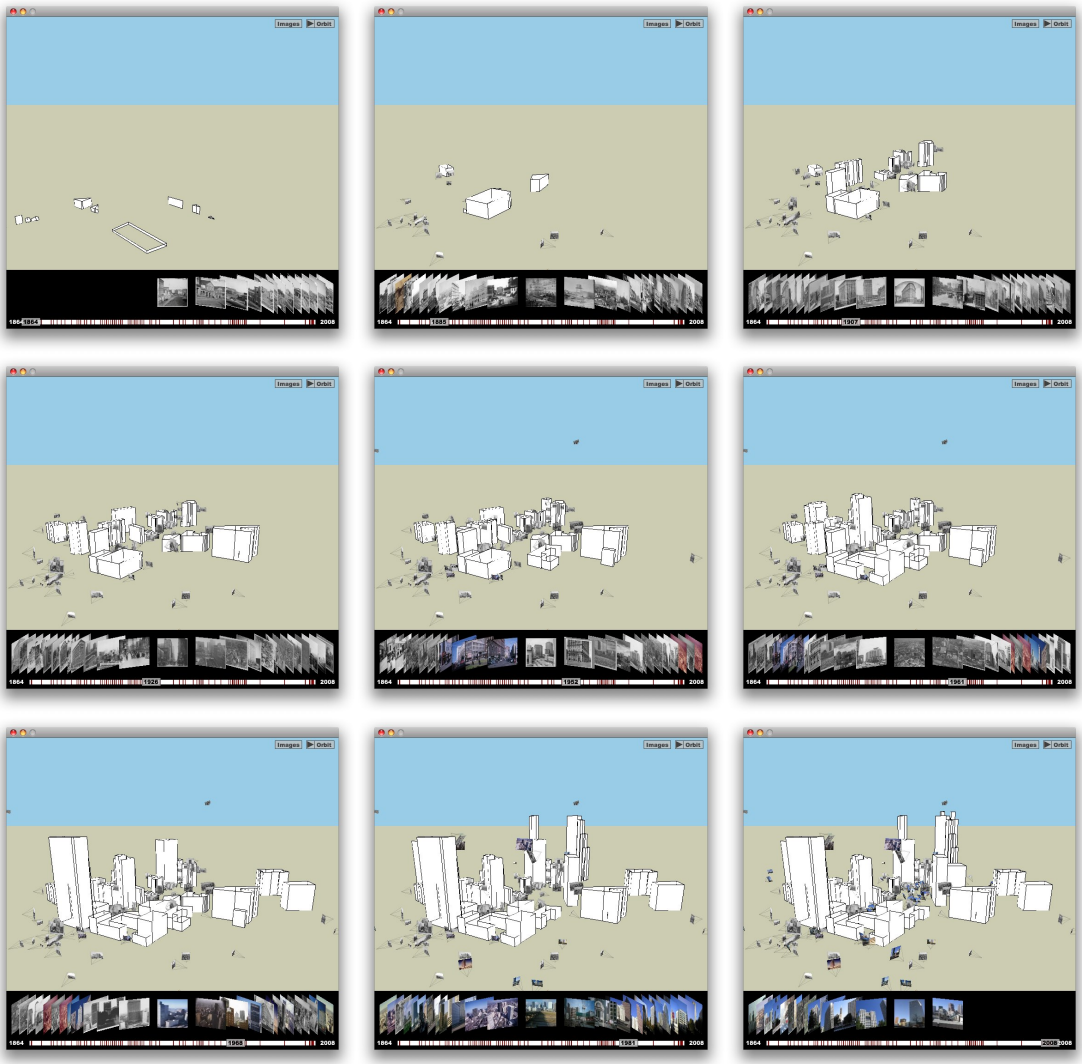


Figure 3: Time-Varying 3D Model. A user can drag a time-slider to see the 3D model of the city at any point in time. As the user does so, buildings rise and fall and images flicker in and out of existence to reflect the changes that have taken place over time.

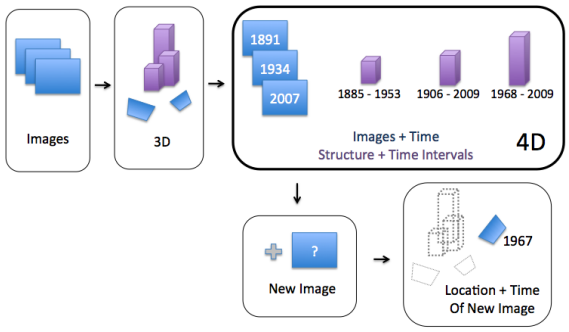


Figure 4: Problem Overview. This block diagram illustrates the basic steps in constructing a 4D model from images. Once we have constructed such a model, we can also use it to determine the location and date of new images.

next, we are able to register these 1864 photographs to the same coordinate frame of our modern photographs which are easily referenced to a global coordinate system via maps or GPS.

Finally, recovering the dates of historical images is an important task still performed by humans that has gone largely unaddressed as a computer vision problem. In this dissertation, we present 4D city models as a solution to the problem of automatic image dating, and we demonstrate the effectiveness of 4D models in recovering dates for images of Atlanta and Manhattan.

1.3 Problem Formulation

1.3.1 Temporal Inference Problem

The problem we are proposing to solve in this dissertation is that of temporal inference from a set of photographs of a scene which changes over time. For every image, we must recover a date and time at which the image was captured, and for every object in the scene, we must recover a beginning and end date (and time) describing when this object existed (see Figure 3). Though we concentrate on historical photographs of cities changing over time, the methods we develop are general. Thus, we present the following general problem formulation involving a set of objects under observation over time:

- A set of m objects $O = \{O_i | i \in 1 \ldots m\}$, where each object has an associated time

interval (a_i, b_i).

- A set of n observations $Z = \{Z_j | j \in 1 \ldots n\}$, where each observation has an associated time t_j.
- Each observation Z_j consists of K_j measurements $U = \{U_{jk} | k \in 1 \ldots K_j\}$
- A set of n correspondence vectors $J_j = \{J_{jk} | k \in 1 \ldots K_j\}$ linking each measurement to a specific object.

The goal of *temporal inference* is to recover the temporal parameters t_j and (a_i, b_i), given the observations, measurements, and correspondences.

1.3.2 Structure from Motion

We formulate the problem of 4D model construction in the context of a Structure from Motion (SfM) framework. Structure from motion is the process of recovering the 3D geometry of a scene (structure) as well as the internal and external camera parameters (motion) associated with a set of images of a scene. Structure from motion is a well-studied problem (Hartley and Zisserman, 2000; Faugeras and Luong, 2001) and in fact, a number of recent methods have applied SfM to large internet image collections (Snavely et al., 2008; L. Lazebnik and Ponce, 2006), mostly to reconstruct famous landmarks and other well-photographed locations, and all ignoring issues of changes over time. Interestingly, recent work in civil engineering *has* begun to use SfM to track 3D changes in building sites over time and compare them to planned models of the same site (Golparvar-Fard et al., 2009).

In this SfM setting, we can begin to make explicit statements about the terms in the above problem formulation. We derive our observations Z from a set of images I by extracting interest points that we treat as 2D measurements K on 3D points in the scene. In this work, we will sometimes treat the 3D points themselves as our objects O, while at other times, we group these 3D points into higher-level geometric structures such as 3D buildings. Our framework applies equally well to other geometric primitives, for example

line-based 3D reconstructions (Schindler and Dellaert, 2006), though we do not explore this case here.

We frame the temporal inference problem in an SfM setting for several reasons. Most importantly, SfM is the necessary first step in converting a set of images into a 4D model. Once we recover a 3D model via SfM, time can be added later to create the 4D model. Second, the problem of simultaneously estimating a set of dates for images, and a set of date intervals for structure elements is *itself* a 1D temporal-structure from temporal-motion problem. Third, to perform temporal inference on images, we use the very same correspondences (see Figure 5) fed into SfM to construct a matrix describing the evidence about each structural element in each image – a modification of the traditional visibility matrix.

In what follows, we describe a temporal inference framework for reasoning about time in images, which comes in two flavors: Relative Temporal Inference, and Absolute Temporal Inference. In the relative case, we are solely interested in the temporal *order* of a set of images (and associated structures), while in the absolute case, we want an exact time and date for every image (and an exact time interval for every structure). Relative Temporal Inference is covered in detail in Chapter 4, while Absolute Temporal Inference is introduced in Chapter 5 and later expanded into a probabilistic temporal inference framework in Chapter 6. Both methods are built around the same underlying framework, which we describe here.

1.3.3 Temporal Ordering: Relative Temporal Inference

To perform relative temporal inference, we must come up with a feasible ordering of the images based on the objects visible and observed in each image. For each image, we form a ternary *visibility vector* V_i where:

- $V_{ij} = 1$ indicates that there is evidence that object O_i existed at the time observation Z_j was taken.

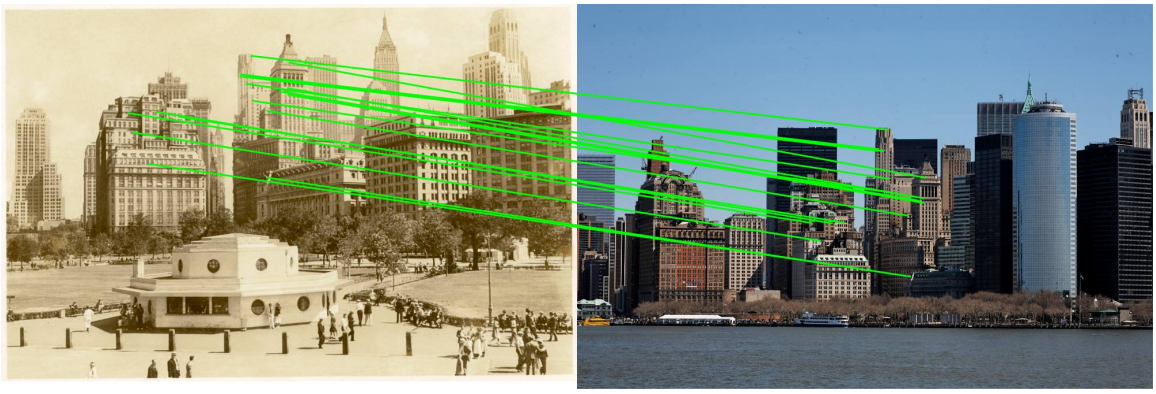


Figure 5: Point Correspondences. To construct 4D city models, we adopt a structure from motion approach which relies on identifying corresponding points in images from different historical periods. Here an image from 1937 is automatically matched to an image from 2009, resulting in 20 correspondences. *Photos provided by New York Public Library (left) and Eric Firley (right).*

- $V_{ij} = 0$ indicates that object O_i was not observed at the time observation Z_j was taken.
- $V_{ij} = -1$ indicates that there is evidence that object O_i did not exist at the time observation Z_j was taken.

We place each visibility vector V_i into a visibility matrix V and reason about the ordering of the columns of this matrix, since each column represents the set of observations about a given image. Rather than explicitly model the time interval (a_i, b_i) for each object, we observe that a given object should never "exist, not exist, then exist again" in the correct ordering of images. With this single constraint applied to all rows of the visibility matrix V, we are able to arrive at feasible orderings of the images by a stochastic greedy local search method. See Chapter 4 for a detailed treatment of relative temporal inference.

1.3.4 Incorporating Image Dates: Absolute Temporal Inference

To convert a 3D reconstruction into a useful 4D model, we are concerned with recovering an absolute date for each camera and an absolute time interval for each structure element in the scene. By absolute date, we mean a specific year, month, and day (up to some desired level of precision) on the standard Gregorian calendar. This is in contrast to the relative dates and time orderings described above.

In this case, time is a continuous variable, and we must modify our algorithms to directly estimate continuous values for each time parameter in the 4D model. At the same time, this gives us the opportunity to incorporate known dates or estimated dates of images directly into the temporal inference process. For example, knowing that a given image was taken in "March 1968", while another was taken "Circa 1950" is quite valuable even if these dates are not entirely precise. In addition, we propose an appearance-matching method of estimating the absolute date of an image by comparing it to a database of labeled images.

We deal with time as a continuous variable by introducing a continuous optimization framework for temporal inference which estimates time parameters of a 4D model by simultaneously minimizing an error related to the observations of structure in each image, while taking into account the known or estimated dates of individual images.

1.3.5 A Probabilistic Framework for Temporal Inference

We finally present a Bayesian framework to find the maximum a posteriori temporal parameters for a scene, and to find a distribution over temporal parameters in a Markov Chain Monte Carlo framework. Fully automated structure from motion methods are used, and we introduce a method to segment large point clouds into objects about which to reason.

A key advantage of this method is that it treats the time-varying geometry of the scene itself as the occlusion geometry used to predict the visibility of objects, given a set of temporal parameters. The framework successfully incorporates uncertain date information, while at the same time making use of probabilistic equivalents of the visibility constraints in the original temporal ordering formulation.

We present results on two large-scale historical reconstructions, one of Atlanta, Georgia and another of Manhattan, New York, as well as on synthetic data which allows us to characterize the performance of this method with respect to ground truth in a number of experiments.

1.4 A History of Temporal Inference

The problem of temporal inference we present here is new in the field of computer vision, but it has analogues in other fields. For example, there are surprising parallels between the problem of dating a historical photograph and determining the age of fossils in the era before carbon dating. In famed geologist Charles Lyell's seminal 1830 book "Principles of Geology," he proposed a new statistical technique for temporally ordering fossils of the Tertiary strata (the most recent 65 million years). In what follows, different fossilized species in geologic strata are represented as different beans in the bag. As reported in Stephen Jay Gould's 1987 book "Time's Arrow, Time's Cycle, pages 160-161:

> The grand beanmaster now sets us a problem. He took an x-ray of the bag every six hours during the last day, but he forgot to mark the times on the negatives, and he wants us to arrange the four photos... in proper temporal order. He is also willing to give us the bag as now constituted at day's end. How can we proceed?
>
> ...Lyell then proposes his statistical criterion. We cannot know when any particular bean entered the bag, but we can make a list of all signatures in the bag as now constituted. We can then study the beanmaster's four photos and tabulate the 1000 signatures in each. The longer any bean is in the bag, the greater its chance of removal... Thus, the more recently any bean entered, the greater the chance that it still resides in the bag. Lyell exclaims triumphantly that we need only tabulate, for each photo of the bag at a previous time, the percentage of beans in the bag that remain at day's end. The higher the proportion of current beans, the younger the photo.

The problem of dating a set of fossils found in a common geologic stratum is nearly identical to the task we confront in performing temporal inference on images if we substitute

buildings for fossils and photographs for geologic strata. However, we have several advantages and therefore we adopt a different technique to the one described in the preceding passage. First, some of our images come with dates attached, even if they are approximate. Second, we can exploit our knowledge of how 3D points project down to 2D points in photographs, and of how 3D objects occlude each other, in order to attack the problem in a more principled way than the statistical solution to temporal inference proposed by Lyell.

1.5 Dissertation Overview

This dissertation begins by discussing what 4D city models are and how to interact with them, then moves on to methods for relative and absolute temporal inference, and concludes with a discussion of the contributions and limitations of the presented methods.

Chapter 2 defines a 4D city model and gives examples of 4D city models we have constructed using the techniques presented in this dissertation. We also discuss a method to interactively build 4D city models using manual point correspondences.

Chapter 3 describes the ways in which 4D city models can be used to interactively display both images and time-varying 3D models. We also describe how to make use of 4D city models in an image-based rendering setting to produce visualizations of change over time.

In Chapter 4, we present a method for inferring the temporal order of images from 3D structure. We introduce the visibility matrix and show how it can be used in a constraint satisfaction setting to solve the temporal ordering problem. We rely on manual point correspondences for 3D reconstruction.

In Chapter 5, we incorporate the notion of real image dates and show how the constraint satisfaction problem is transformed into a continuous optimization problem for solving temporal inference.

In Chapter 6, we introduce a probabilistic temporal inference framework which captures the visibility constraints of the temporal ordering method and properly incorporates

uncertain knowledge about image dates. We also show that the point clouds resulting from automatic structure from motion methods may be segmented into buildings for temporal inference. In addition, this chapter shows results of the probabilistic temporal inference framework on large-scale real and synthetic scenes.

Finally, in Chapter 7, we discuss limitations of the presented approach. Specifically, we discuss the difficult problem of detecting corresponding feature in images taken at different historical times, and we present results which characterize the reliability of matching SIFT features over time. We also reiterate the contributions of this dissertation, and discuss directions for future work.

Chapter II

4D CITY MODELS

A 4D city model is a useful tool for organizing photographs and understanding their temporal and spatial context. In this chapter, we define the concept of a 4D city model and we show several examples to demonstrate what they look like and why we want to build them. We briefly detail an interactive method of constructing such 4D city models, leaving to later chapters the issue of how to go about automating this task.

2.1 What is a 4D City Model?

A 4D city model is a time-varying 3D model of a city. It consists of a number of 3D geometric primitives (such as points, lines, triangles, polygons), each with an associated time interval. The geometry of the scene changes over time only due to primitives beginning and ceasing to exist, but the geometry itself never moves through space. Thus, a 4D city model could contain a 3D point that lasts for five minutes, or a polygonal model of a building that exists for 100 years. Note that we first define 4D city models without considering the types of algorithms we will use to build them. Later, we will see that such models can be built either manually or automatically.

Essential to the idea of 4D cities, in this work, is the concept of a set of photographs geometrically registered to the geometry of the scene, and taken at different points in time. For every image, we must know the 3D pose of the camera, internal parameters such as focal length, and the date and time at which the photograph was taken. Though time-varying 3D models can exist independently of any set of photographs, when we talk about a 4D city model in this work, we are assuming that such a set of photographs is present for two reasons. Firstly, we see one of the most important functions of a 4D city model as organizing the historical photographic record of a given city, and enabling new ways of

Figure 6: Example View of 4D City Model of Atlanta in 1971.

understanding historical photographs in their spatial and temporal context. Second, one of the claims of this work is that such a set of photographs has enough information to enable the construction of 4D city models from images alone.

2.2 4D City Examples

In addition to the theory and methods underlying 4D city creation, one of the main contributions of this work is the actual 4D city models which have been produced as a result. Through the methods developed in this dissertation, we have created functioning 4D city

Figure 7: Example View of 4D City Model of Atlanta. Civil War photograph from 1864.

models that have not only served as tools of organization and interaction for photographs, but tools of discovery and analysis of historical details as well. We briefly discuss and show visual examples of 4D city models for Atlanta, Manhattan, and Seoul below.

2.2.1 Atlanta

The city for which we have constructed the most complete 4D model is Atlanta, Georgia. Through a relationship with the Atlanta History Center, we were able to acquire a collection of images spanning 1864 to the present. Using methods described in this dissertation, we

Figure 8: Example View of 4D City Model of Atlanta in 1907. The Candler Building, highlighted in red, was built in 1906 and still exists today.

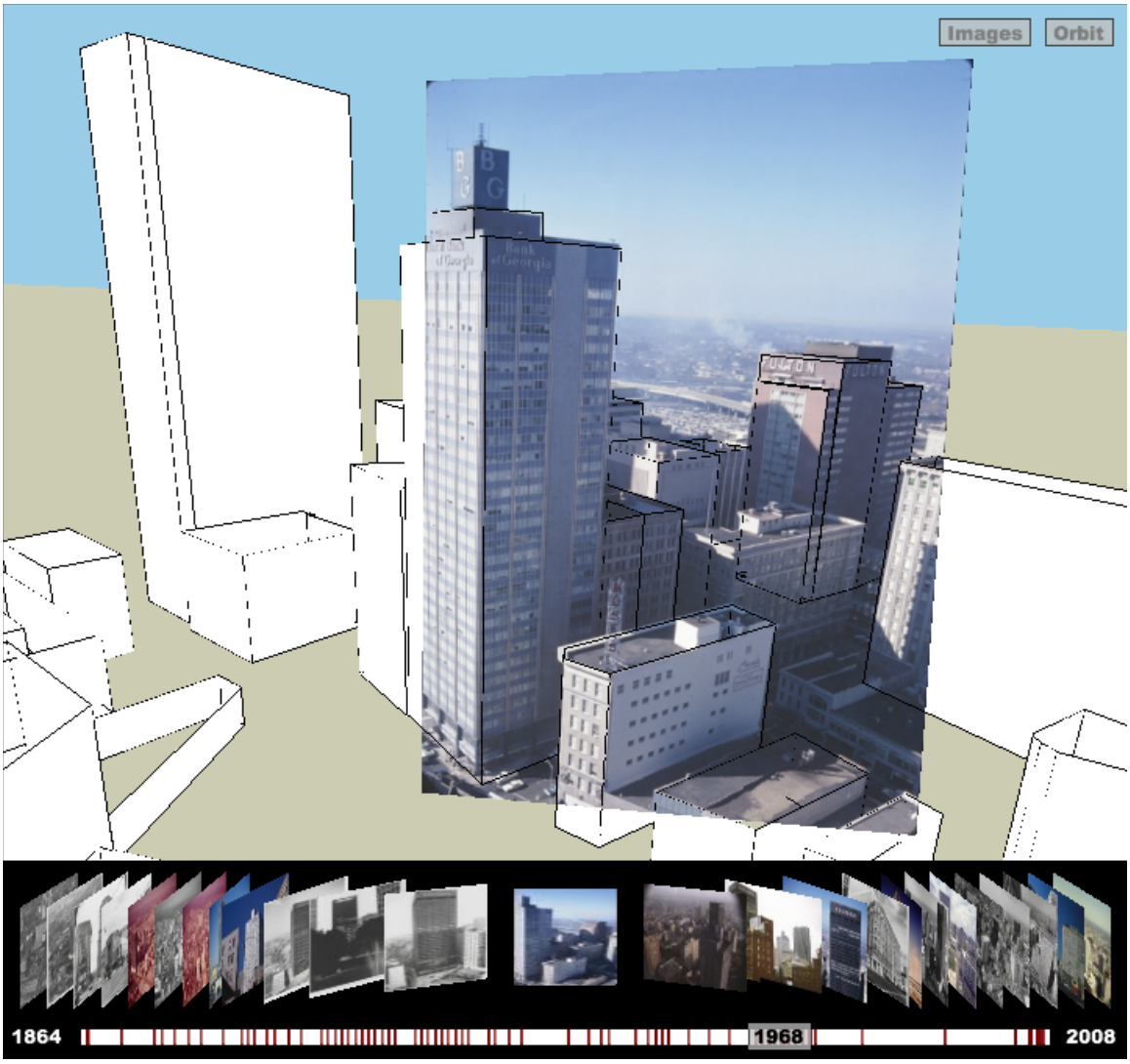


Figure 9: Example View of 4D City Model of Atlanta in 1968. Even buildings completely or partially outside the field of view of this 1968 photograph are shown in the model because they existed at the time the photo was taken.

Figure 10: Example View of 4D City Model of Atlanta in 1969. By extending beyond the edges of the photograph, the model provides both spatial and temporal context for the image.

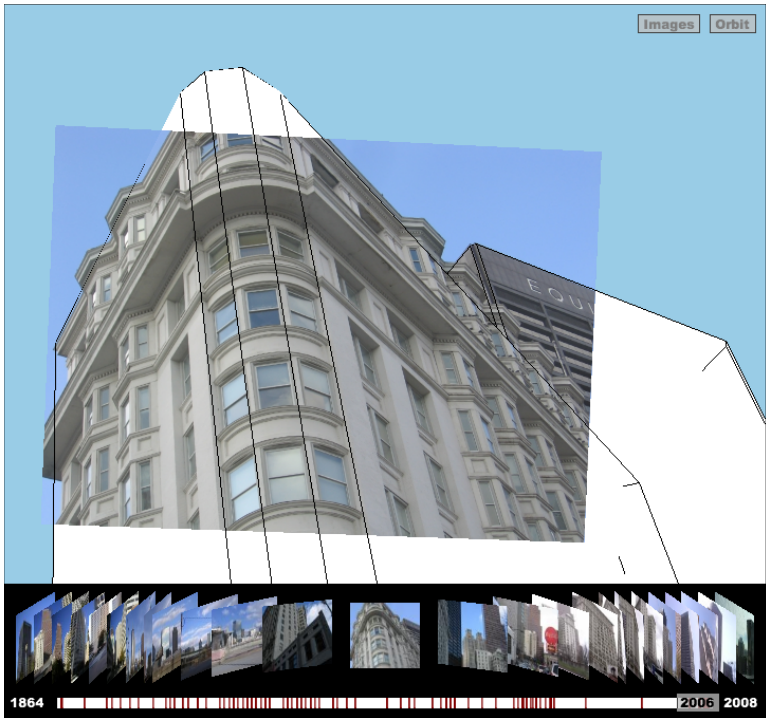


Figure 11: Example View of 4D City Model of Atlanta in 2006, showing parts of the Flatiron Building (1897) and the Equitable Building (1968).

are able to incorporate over 200 images into a consistent 4D model of downtown Atlanta as it has changed over nearly a century and a half. The oldest photographs in the collection were captured by Civil War photographer George Barnard in 1864 (see Figure 7). None of the buildings pictured in the 1864 photographs still exist, and most were destroyed shortly after the photographs were taken. It is only due to the temporal density of our photo collection that we are able to construct a single 4D city model that spans such a large period of time. This dense sampling in time leads to pairs of photographs always having some common structure through which to link the images geometrically. Figures 8 through 11 show additional views of Atlanta which demonstrate how photographs from various eras are enhanced by the additional spatial and temporal context provided by the 4D model.

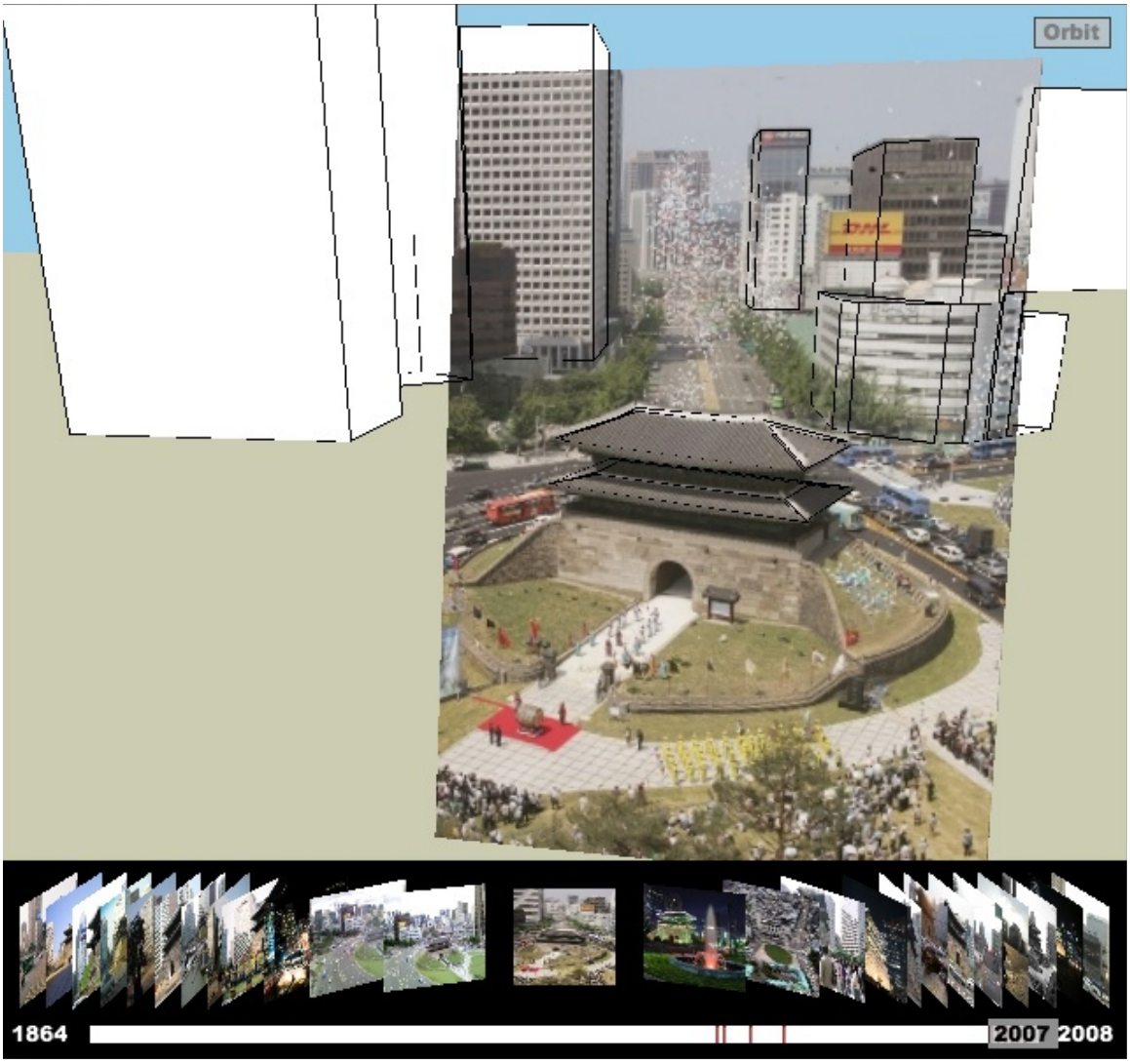


Figure 12: Example View of 4D City Model of Seoul in 2007. The famed Namdaemun gate pictured here was burned down in a 2008 incident.

2.2.2 Seoul

We have also constructed a 4D model of Seoul, South Korea based around the famed Namdaemun gate which was destroyed in a fire in 2008. The gate was originally built in 1398 as the southern gate of the walls of Seoul. By organizing images of the monument into a single 4D model, we can begin to understand how much older it is than every other structure surrounding it. This is just one example of how 4D models can help in preserving cultural heritage and fostering historical understanding.

Figure 13: Example View of 4D City Model of Seoul. Here we see a 1960 image with modern buildings overlaid.

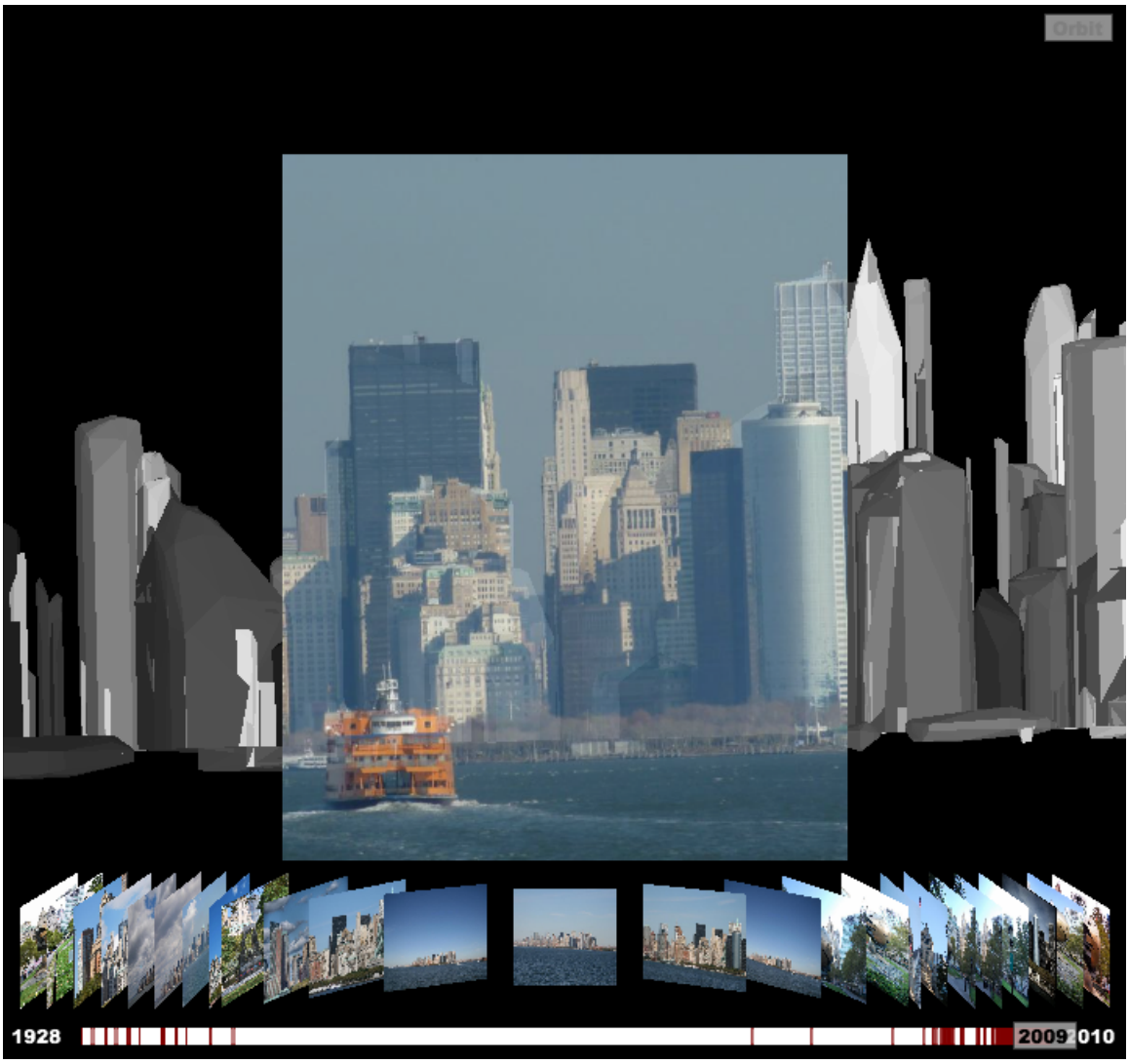


Figure 14: Example View of 4D City Model of Manhattan in 2009, as viewed from the Staten Island Ferry. The 3D building models seen here were constructed automatically from images of Lower Manhattan. *Photo by Flickr user iainh124a (Iain Henderson).*

2.2.3 Manhattan

Our 4D model of Lower Manhattan was constructed completely automatically from over 450 images of this region of New York City. It brings together a collection of historic photographs from the New York Public Library, depicting the 1920s, 1930s, and 1940s, with modern digital images from Flickr depicting the 1980s, 1990s, and 2000s. Many of the images were taken in Battery Park on the southern tip of Manhattan, or from the Staten

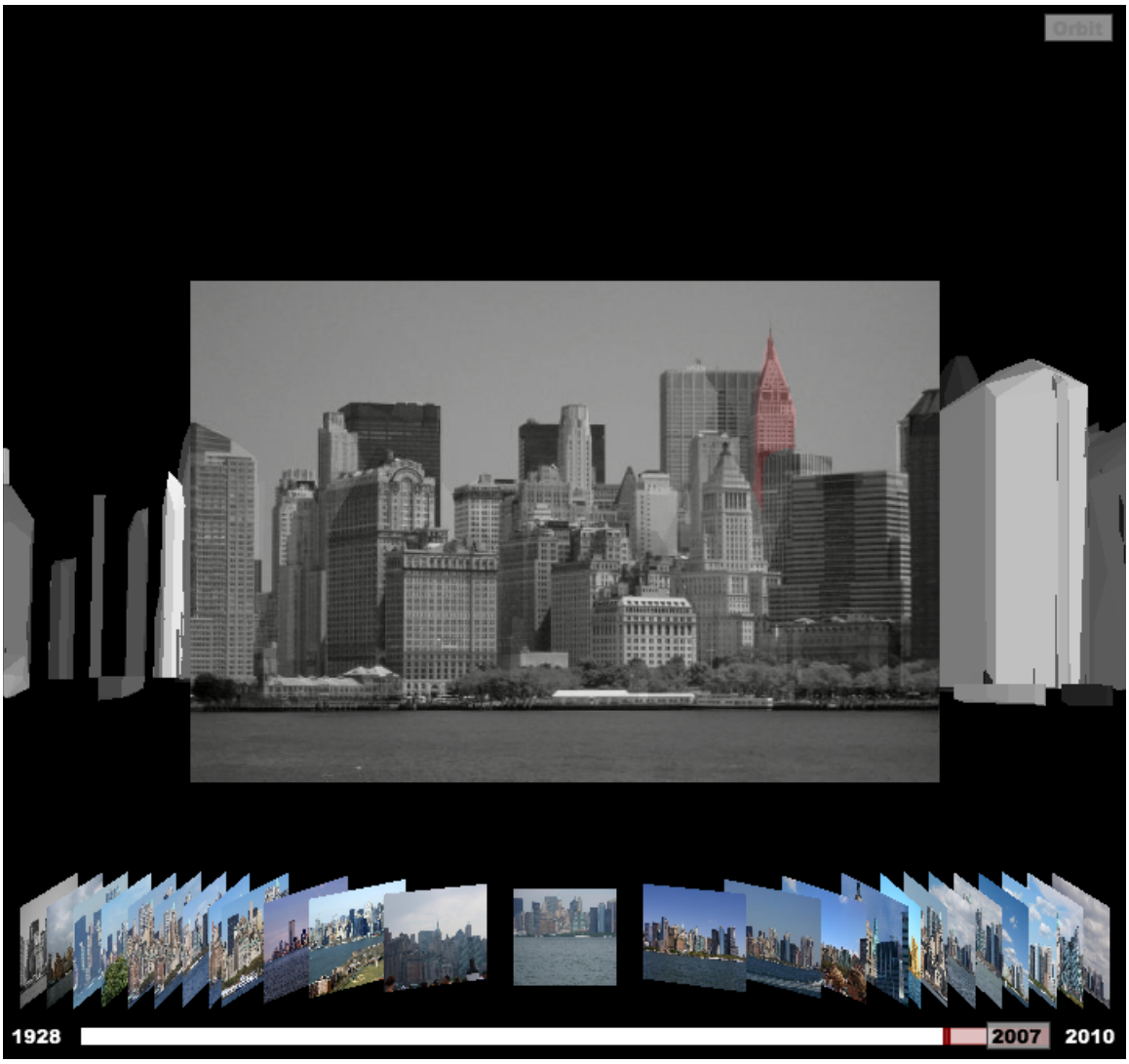


Figure 15: Example View of 4D City Model of Manhattan in 2007. When exploring a 4D city model, a user can select individual buildings, like the one highlighted in red, to find out which photographs depict that building, and when they were captured. *Photo by Ray Kippig*.

Island Ferry. Included in this model are the Twin Towers of the World Trade Center (see Figure 16), as well as *The Sphere*, a sculpture which formerly sat at the base of the Twin Towers and is now on display as a memorial in Battery Park (see Figure 17).

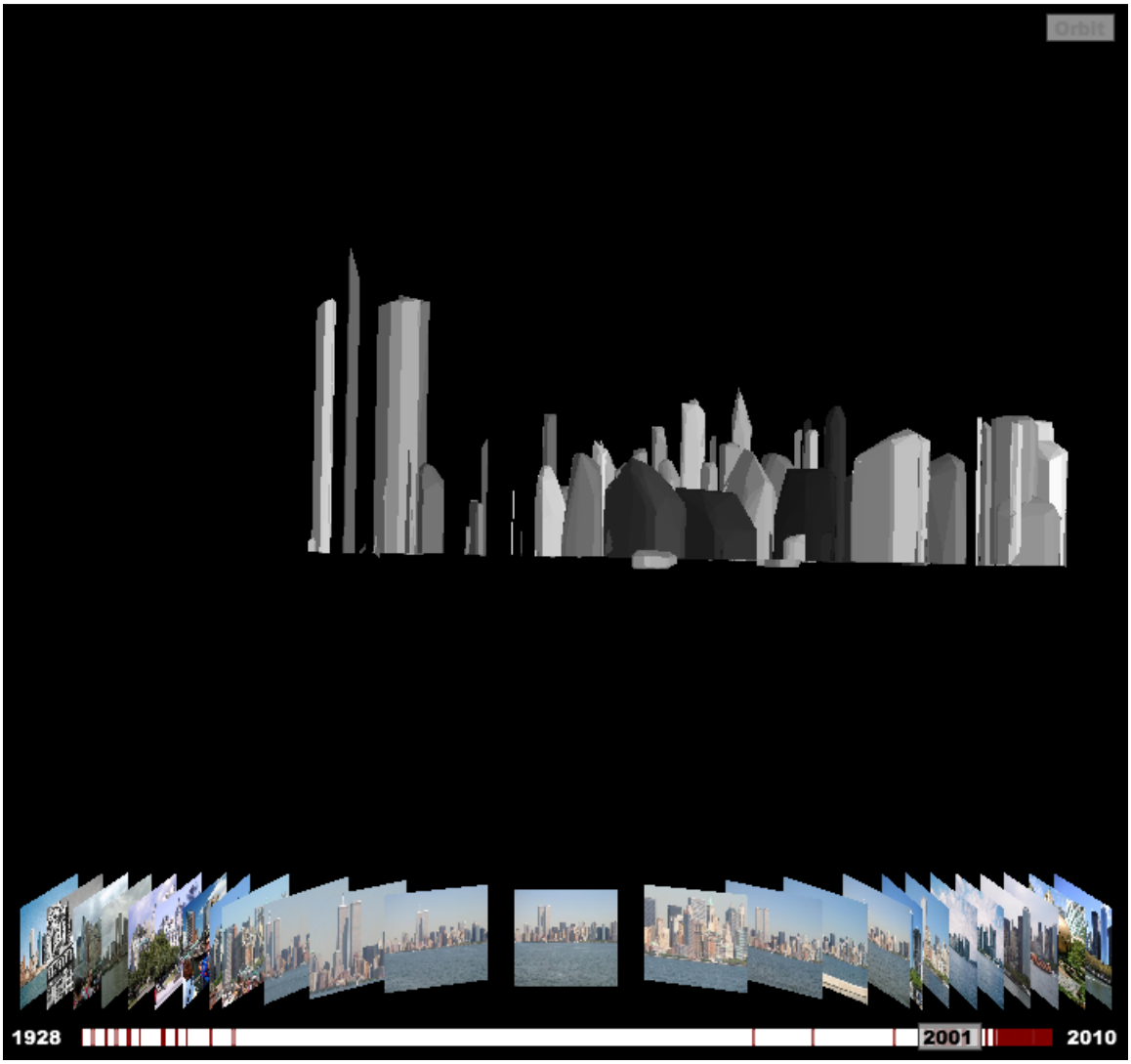


Figure 16: Example View of 4D City Model of Manhattan in 2001. On the left, we see the Twin Towers of the World Trade Center.

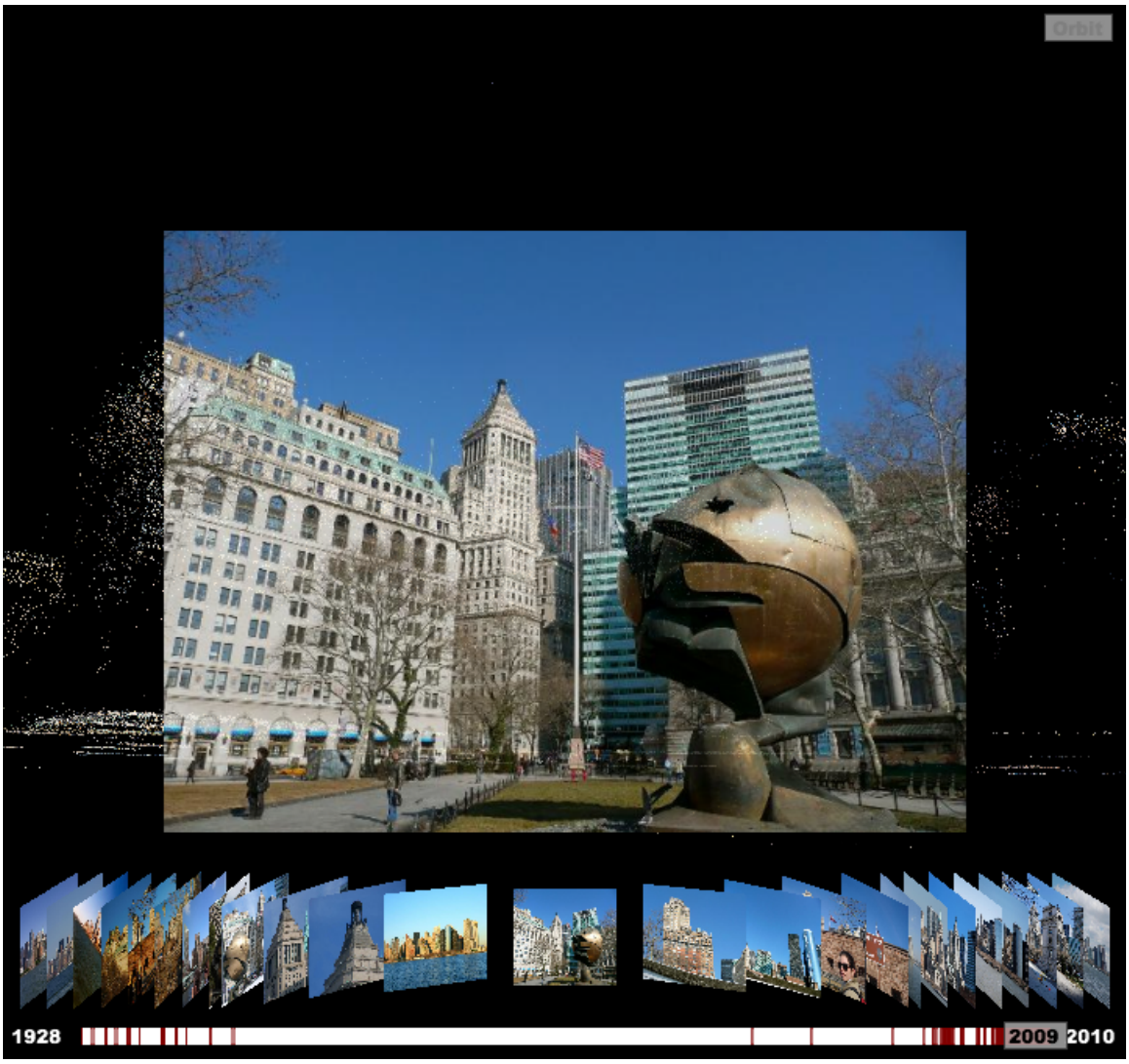


Figure 17: Example View of 4D City Model of Manhattan in 2009. The sculpture pictured here formerly sat at the base of the Twin Towers and now resides in Battery Park as a memorial. *Photo by Rachael McCurdy.*

2.3 Constructing 4D City Models

4D city models, like the ones pictured above, can be constructed either automatically or interactively with a user guiding the process. The Atlanta and Seoul models pictured above were constructed using an interactive method which we briefly detail here. In contrast, the Manhattan model pictured above was constructed automatically using methods detailed in Chapter 6.

Constructing a 4D city model, at its core, involves identifying corresponding points across multiple images. Though modern computer vision techniques are capable of performing this task automatically for certain image collections, there are distinct advantages to allowing users to manually specify corresponding points with the help of a user interface designed for this task.

Primary among the advantages of interactively constructing 4D models is that humans can identify corresponding points despite enormous changes in appearance that takes place over time. For the case of Atlanta, by using manual point correspondences we are able to produce a user-constructed 4D model spanning the dates 1864 to 2008 (across 212 images). See Figures 7 through 13 for examples of models constructed in such a way. In contrast, we find that automated correspondences only produce a 4D model of Atlanta spanning 1956 to 1975 (across 102 images) due to the inability to detect corresponding SIFT features across the entire database of images. This problem is detailed in Chapter 7.

A second advantage to putting humans in the loop is that it allows the creation of simplified solid building geometry. Though we present several methods of automatically segmenting and triangulating the point cloud resulting from automated SfM methods, the resulting building models can be incomplete and noisy, and may split or merge buildings incorrectly. Interactive modeling methods avoid this problem.

Therefore, we have created a 4D city construction tools which consists of an interface:

- to specify corresponding points between two images,
- to define a building by joining a series of points,
- and to specify a date for each image and a time interval for each building in the scene.

To recover camera parameters and 3D scene geometry, we use a custom bundle adjuster based on Levenberg-Marquardt within an automatic differentiation framework (Griewank, 1989). After specifying point correspondences, a user can choose to optimize structure parameters, optimize camera parameters, or optimize all parameters simultaneously using

Figure 18: 4D City Models. These models were constructed using manual point correspondences which were specified with a user interface that makes it easy to create 3D buildings. This tool was used to construct 4D models of Atlanta, Georgia (top) and Seoul, Korea (bottom).

bundle adjustment. This is the only operation we require for 3D reconstruction, as we depend upon user input to initialize cameras and points to reasonable values.

Model construction begins with an initial pair of images in which the user specifies correspondences for a specific set of 4 points which define the origin, scale, and coordinate axes of the world. Additional prior terms are added to the bundle adjustment in order to constrain these points during subsequent optimization. In addition to these constrained points, the user specifies a height for the ground plane which is used in the interactive viewer. Knowledge of the direction of gravity and a ground plane enables buildings to be defined by simply specifying an ordered set of points along the roof of a building which are to be connected into a polygon and extruded to the ground. Such simple building models are useful both for modeling occlusions and during subsequent user interaction with a 4D model to determine when a user has clicked on a given building in any image (see Chapter 3).

Once the geometric parameters of the initial images and points have been solved, the user alternates between adding additional 3D points to the model (initialized by back-projection into existing images) and adding additional images (initialized with the pose of an existing camera in the reconstruction). This procedure enabled the creation of the models for Atlanta, Georgia and Seoul, South Korea as depicted in Figure 18.

In the next chapter, we discuss methods of interacting with and visualizing these 4D city models.

Chapter III

VISUALIZING AND INTERACTING WITH 4D CITIES

One of the primary motivations behind constructing 4D city models is that they enable new ways of interacting with both historical and modern imagery. In this chapter, we describe the new interaction techniques we have developed for viewing and exploring 4D city models. We provide illustrative examples of how 4D city models can be used as a tool of historical discovery. Finally, we discuss techniques for visualizing 4D city models in a non-interactive manner.

3.1 Viewing 4D Cities

Historical and modern images are currently dispersed across a wide variety of online sites, including Flickr, Picasa, the Library of Congress, and numerous smaller collections at various universities, historical societies, and other institutions such as The Atlanta History Center, The New York Public Library, and the Charles W. Cushman Photograph Collection at Indiana University. The goals of a 4D city viewer include:

- to bring together historical and modern photos from a variety of sources
- to place these photos in both their *spatial* and *temporal* context
- to allow a user to see how a whole city, a specific building, or a specific view changed over time

To enable this interaction we require precisely the type of representation outlined above: a set of images with known pose, calibration, and date, and a collection of buildings with known 3D geometry, date of construction, and (if applicable) date of demolition. A list of which buildings are observed in which images is also necessary to take full advantage

of our 4D model interaction methods. For the purpose of interacting with a 4D model, as long as we have all these pieces of information, then how we acquire the model is not important (and any of the interactive or automatic methods described in this dissertation may be used).

Given a 4D city model, we define a number of ways for the user to interact with this model, which we outline briefly below. To see examples of the visual elements of the user interface described here, refer to Figure 19 or any of the other figures in this chapter and the previous chapter.

Timeline The primary novelty of this interface is a timeline which lets the user set the current time at which to view the model. Tick marks along the timeline indicate dates at which photographs exist in the model, and corresponding thumbnail images are arrayed along the timeline. As the user drags the time slider back and forth, both the 3D model of the city and the displayed images change to reflect the current date.

3D View The user sees the entire 3D city model from an overhead viewpoint and is able to orbit around the city with a mouse. Along with 3D building models, this view also shows images floating in space at the position and orientation of the associated camera. The user may select any of these floating images, or the images along the timeline, to view the model from the viewpoint of any individual photograph. When any photo is selected, the date of the time slider changes to the date on which the image was captured, thus changing the displayed 3D buildings as well.

Image View From the viewpoint of any image, the photograph itself is overlaid on the 3D model such that the buildings present in the image are clearly outlined by the corresponding 3D building models underneath. The user can rotate this viewpoint with the mouse to look around the scene and see which buildings were present at the time of the photograph, *despite the fact that they lie outside the field of view of the camera*. In addition, from the

same image viewpoint, the user can drag the time slider long the timeline to show what the current viewpoint would have looked like in a different era.

Building Selection The user may click on any building, whether from the orbiting 3D view or image view, in order to highlight it. When this happens, the images on the timeline and in the 3D view are filtered down to only those images that view the highlighted building. In addition, the area of the timeline between the beginning and end dates of the selected building are highlighted as well, and the set of remaining date tick marks on the timeline gives an indication of the periods from which we have images that observed this building.

Note that if any tick mark lies outside the highlighted region of the timeline for this building, we know there is an inconsistency between this image date and the time interval for the selected building. Similarly, if from the viewpoint and date of a specific image a building in the image is not being shown by the 3D model (or vice versa), we know there is a temporal inconsistency in the model as well. These two cases illustrate that, even without any automated dating mechanism, just relating all the images to a 4D city model and visualizing the result is a powerful tool for ensuring consistency between photographic dates and historical building records.

3.2 Historical Discovery

We claim that 4D city models may be used as a tool of historical discovery, and here we provide two examples of how a user might make such discoveries by interacting with the model.

Buildings change both visually and physically over time, which can make it difficult to recognize the same building in both modern and historical photos. This is further complicated by the fact that the spatial context of the building may change as new, taller buildings are built around it. A 4D model makes it easy to find the same building in historical and modern photos simply by selecting the building, which becomes highlighted in all photos

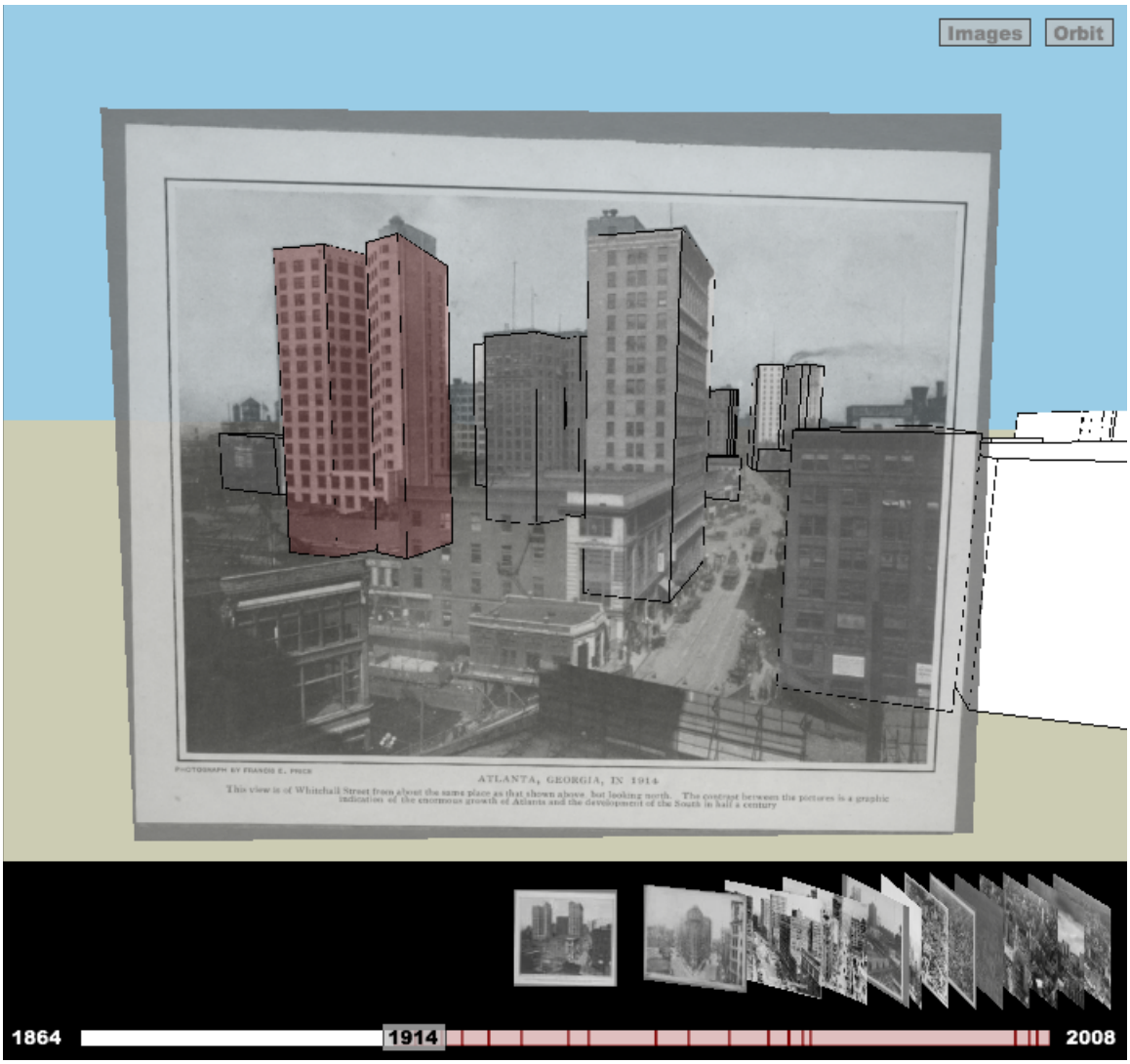


Figure 19: The Fourth National Bank Building, Atlanta, 1914 (highlighted in red). This is the earliest image we have of this building.

Figure 20: The Metropolitan, Atlanta, 2007. A modern image showing the same building as the previous figure (formerly the Fourth National Bank Building), identifiable using the 4D model, despite large changes in appearance.

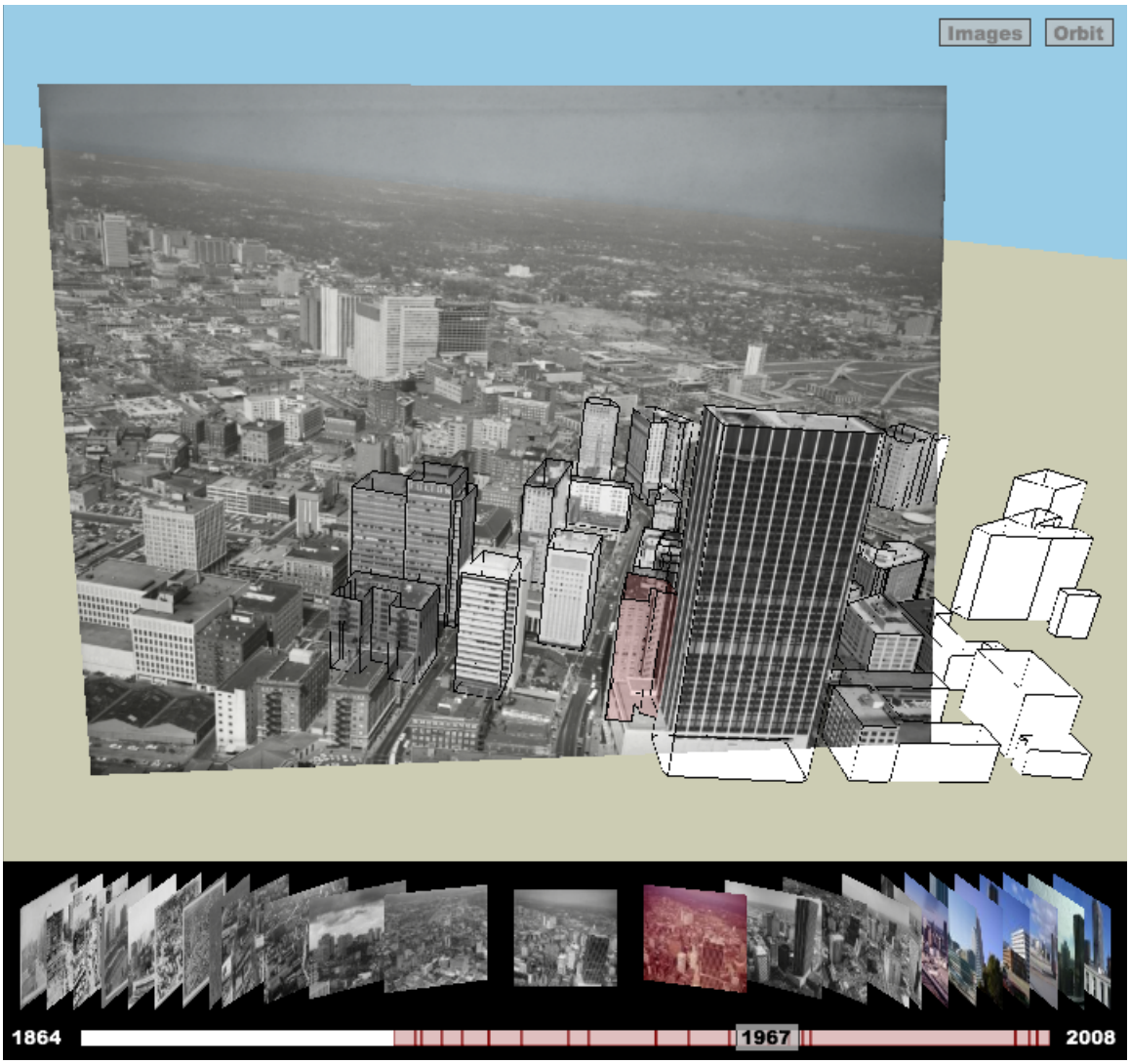


Figure 21: The Fourth National Bank Building, Atlanta, 1967. The last image depicting the building with its original facade.

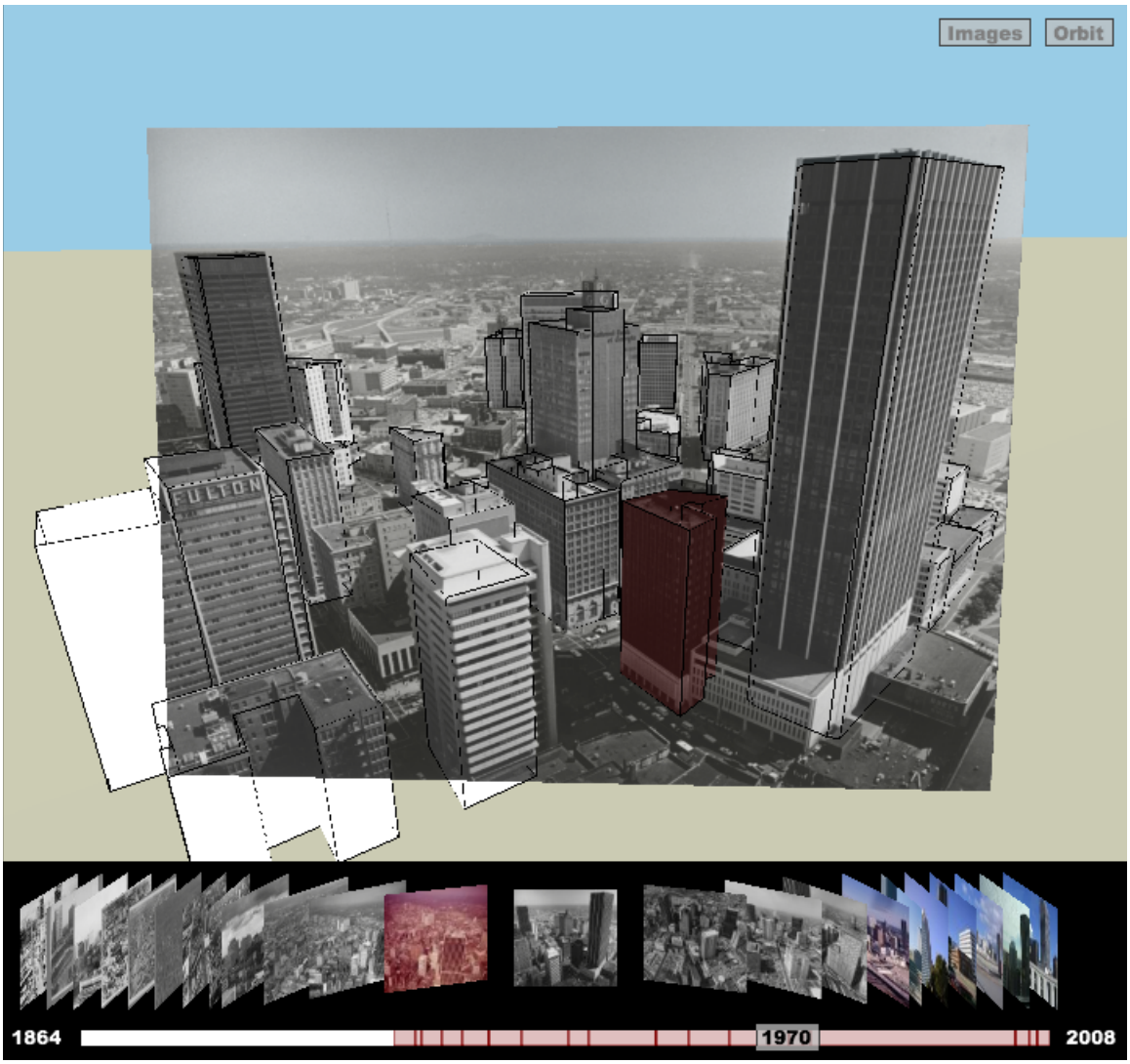


Figure 22: The Fourth National Bank Building, Atlanta, 1970. The first image depicting the building with its new facade.

in which it is visible.

As an example, using a 4D model of Atlanta, we are able to discover that The Fourth National Bank Building (1914) and the Metropolitan (2007), two buildings which differ vastly in appearance, are in fact the same exact building. The Fourth National Bank Building in downtown Atlanta appears to have a light stone facade in the 1914 image depicted in Figure 21. Meanwhile, the Metropolitan, a building in downtown Atlanta in 2007, appears to have a black facade decorated with metal strips as shown in Figure 20. The 4D model tells us that, despite the change in appearance, this is the same building depicted in the two images.

Moreover, we can determine when this change occurred by quickly flipping through all images which observe the given building. The building remains highlighted in red as we flip through the images, and we are able to determine that the change occurred between 1967 and 1970. Figures 21 and 22 show the last image depicting the building with its original facade, and the first image depicting the building with its new facade.

Another type of discovery is determining the location from which a particular photograph was taken. Rather than just determining the GPS coordinate of an image, a 4D model can show us that a photographer was standing on top of another building, or looking out the window of another building, when a photograph was captured. For example, we see in Figure 23 a photograph of the downtown Atlanta skyline in 1951. There is a rooftop visible in the image, and one might assume the photo was captured from this same rooftop. When we select the building in question (the old Equitable Building) and back out to a wider view, we see that this is not the case. In Figure 24, the 4D model reveals that the image was taken from the rooftop of a different building (the Hurt Building), not the rooftop visible in the image itself. In addition to the historical significance of finding out where a specific photographer stood over 50 years ago, it tells us that we could go take a photograph from this same viewpoint today because the Hurt Building still exists, despite the fact that the old Equitable Building no longer stands.

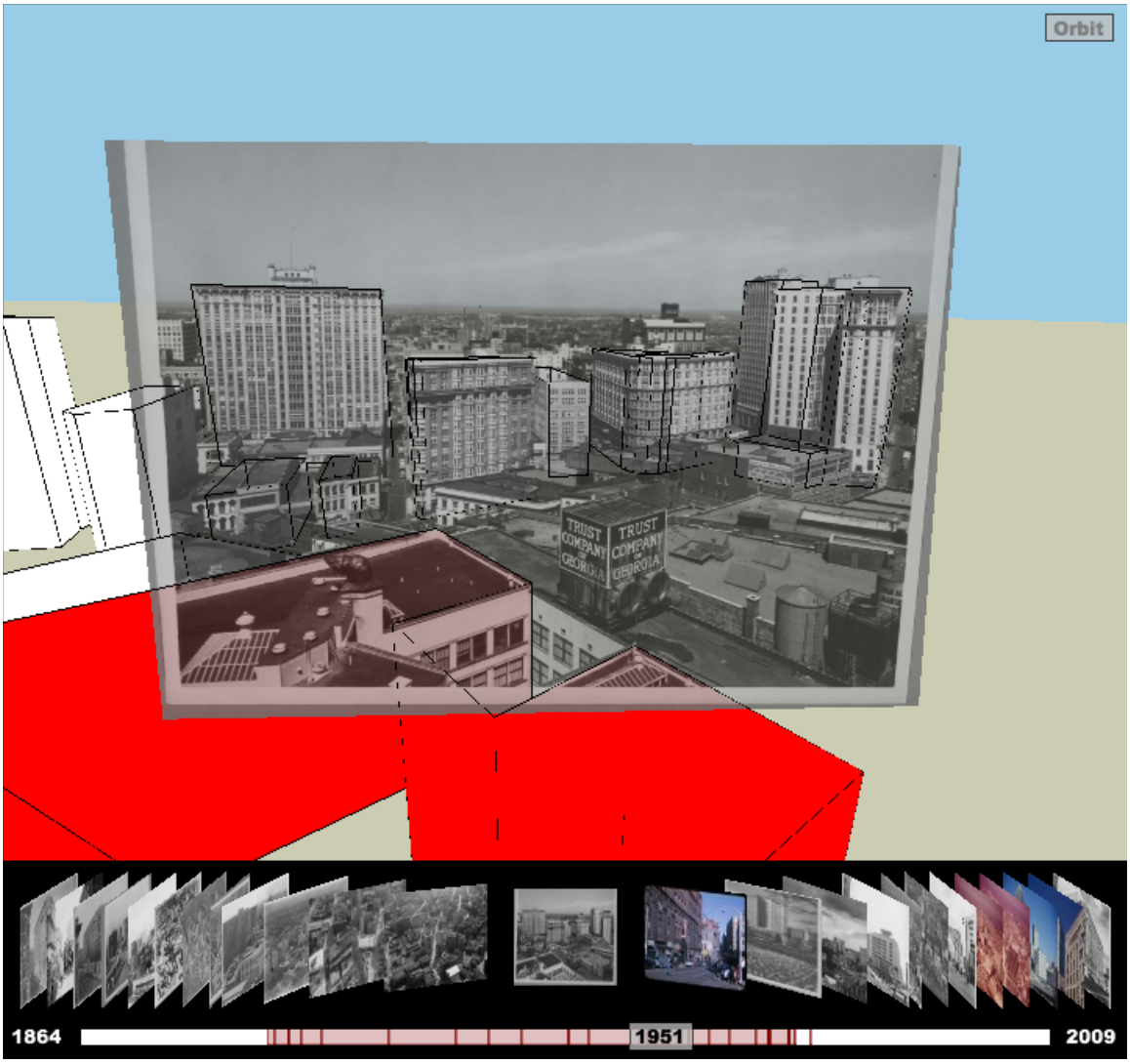


Figure 23: Downtown Atlanta in 1951.

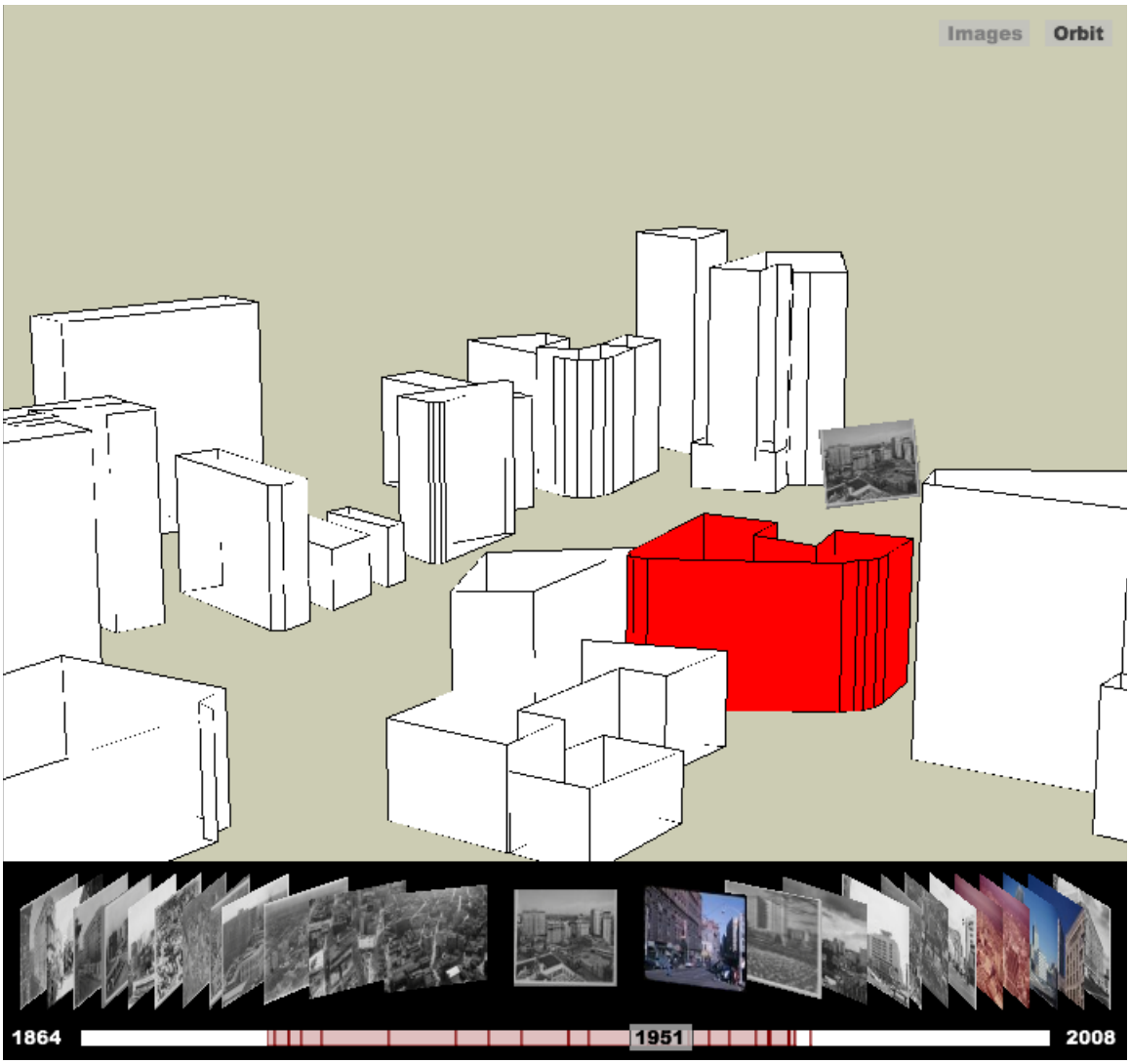


Figure 24: Downtown Atlanta in 1951. The 4D model reveals that the image was taken from the rooftop of a building, though not the rooftop visible in the image itself.

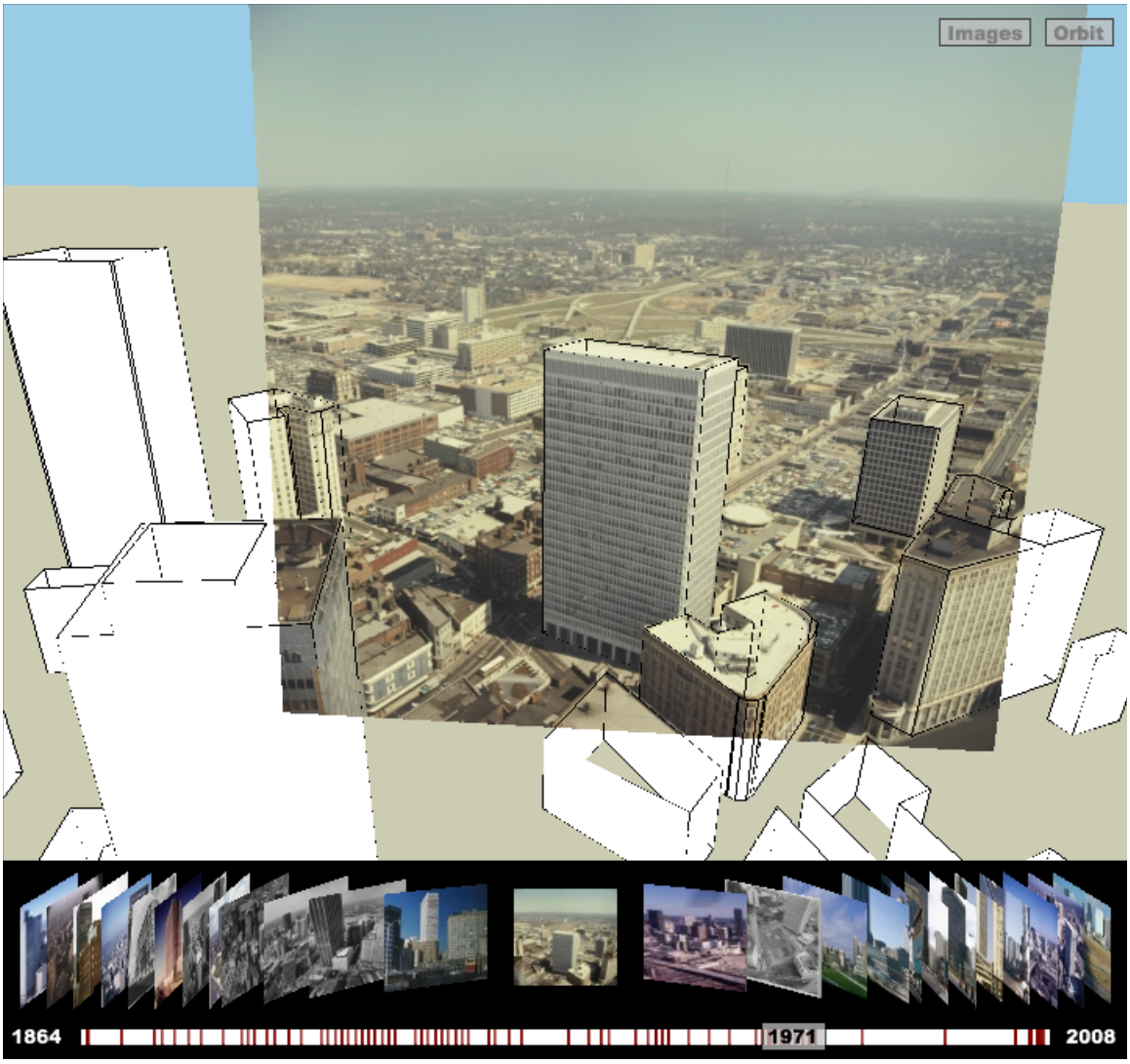


Figure 25: Downtown Atlanta in 1971.

Figure 26: Downtown Atlanta in 1971. What seems at first to be an aerial image was actually captured from the rooftop of the recently finished State of Georgia Building, the tallest in the Southeast United States at the time.

Figures 25 and 26 show an example of an image which one might assume was captured from a helicopter, but was in fact captured from the rooftop of the newly constructed State of Georgia Building, the tallest building in the Southeast United States at the time. For both examples above, there turn out to be other images captured from the same corners of the same rooftops years later, indicating that these locations were quite popular spots for photographing the skyline.

Finally, it is important to note that this type of analysis can be carried out by a user with no expertise, since all the relevant information is captured by the 4D model itself.

3.3 Visualizing 4D Cities

There are several unique visualization techniques that become possible when we have a 4D city model consisting of images taken of the same scene (at different historical dates) registered to time-varying 3D geometry. We focus here on image-based rendering methods, which involve projecting the original images as textures onto the 3D geometry, as distinct from the real-time interactive visualization in the previous section which employs textureless 3D models.

3.3.1 Image-Based Rendering

Often we see a historical image of city that has changed so much that it is difficult to tell exactly where, geographically, the photo was taken. We might be told that a photograph was taken looking North from a given intersection, but without any structures co-existing in the historical and modern day photographs, there is a lack of genuine understanding of the context of the photograph. We can overcome this problem by rendering modern-day buildings in precisely the location they would have appeared had the historical photograph been taken years later.

In Figure 27, we juxtapose different eras in the same photograph, rendering buildings from the 20th century and inserting them into an 1864 photograph of Atlanta. Since we know the internal and external camera parameters for the original 1864 photograph, we can

Figure 27: Visualizing a 4D City Model. We juxtapose different eras in the same photograph, rendering buildings from the 20th century and inserting them into an 1864 photograph of Atlanta. Since we know the internal and external camera parameters for the original 1864 photograph (bottom left), we can render a 3D model of the city from the same viewpoint (bottom right), and pull textures for this 3D model from two other photographs taken in 1966 and 2008. As a result, we get context for the 1864 photograph that is lacking in the original photograph.

Figure 28: Failure of Traditional Image-Based Rendering. If we had only static geometry for the city, rather than time-varying geometry, then traditional image-based rendering techniques would fail by projecting image background onto non-existent 3D geometry. By knowing a date for each image and a time-interval for each building, we avoid this problem.

render a 3D model of the city from the same viewpoint, and pull textures for this 3D model from two other photographs taken in 1966 and 2008. As a result, we get context for the 1864 photograph that is lacking in the original photograph.

The time-varying nature of a 4D model requires a slight change to traditional image-based rendering techniques which assume static geometry (Debevec et al., 1996, 1998). If we had only static geometry for the city, rather than time-varying geometry, then traditional image-based rendering techniques would fail by projecting image background onto non-existent 3D geometry as in Figure 28. By knowing a date for each image and a time-interval for each building, we avoid this problem.

3.3.2 Animated Image Transitions

Another powerful tool to communicate a changing scene is an animated transition between two images taken at different historical times. Here, we use an image morphing (Seitz and Dyer, 1996; Wolberg, 1998) method to transition between two images taken from a similar viewpoint. The known geometry of the scene is used to create a 2D mapping between pixels in the two images as a virtual camera transitions between the known viewpoints of the two original images. This morphing-based method avoids visual holes where no geometry exists (e.g. in the sky).

Figure 29: Animating a transition between two images. We use the known time-varying 3D geometry to morph between two different viewpoint and time-periods. *Photos provided by New York Public Library (left) and Tony Street (right).*

In Figure 29, we show a transition between a 1937 image of Lower Manhattan and one from 2001. Some of the buildings remain the same between the two images, while there are also a large number of new buildings that appear. The transition makes it clear which buildings are new, which buildings remain, and how the two viewpoints are related.

3.4 Conclusion

In this chapter, we have discussed ways of interacting with and visualizing 4D city models once they are built. In the remainder of this dissertation, we will discuss ways to automate the process of 4D city model construction and temporal inference in order to enable these types of interaction and visualization.

Chapter IV

TEMPORAL ORDERING: RELATIVE TEMPORAL INFERENCE

In this chapter, we introduce the visibility reasoning concepts at the root of all the temporal inference methods presented in this dissertation. We show that reasoning about the visibility of 3D structures in images leads to a solution for a temporal ordering of those images in a constraint satisfaction framework. For now, we are only interested in finding an ordering of the images, and we make use of no prior knowledge about the dates of images or buildings.

4.1 Problem

Cameras and skyscrapers have now coexisted for more than a century, allowing us to observe the development of cities over time. We are interested in being able to automatically construct a time-varying 3D model of a city from a large collection of historical images. Such a model would reflect the changing skyline of the city, with buildings created, modified, and destroyed over time. It would also be useful to historians and urban planners both in organizing collections of thousands of images (spatially and temporally) and in generating novel views of historical scenes by interacting with the time-varying model itself.

To extract time-varying 3D models of cities from historical images, we must perform inference about the position of cameras and scene structure in both space and time. Traditional structure from motion (SfM) techniques can be used to deal with the spatial problem, while here we focus on the problem of inferring the temporal ordering for the images as well as a range of dates for which each structural element in the scene persists. We formulate this task as a constraint satisfaction problem (CSP) based on the visibility of structural elements in each image. By treating this problem as a CSP, we can efficiently find a suitable ordering of the images despite the large size of the solution space (factorial in the number

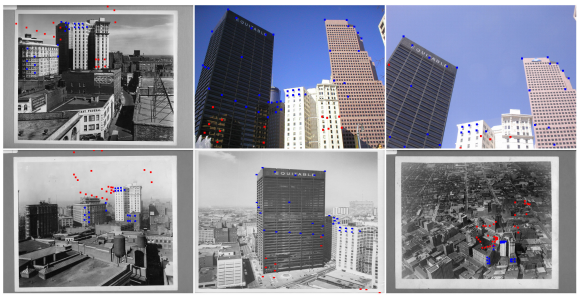


Figure 30: Given an unordered collection of photographs, we infer the temporal ordering of the images by reasoning about the visibility of 3D structure in each image.

of images) and the presence of occlusions.

4.2 Related Work

SfM is now a well-studied problem, and the early stages of our approach proceed very much in the same manner as in (Snavely et al., 2006), recovering calibrated cameras and the 3D point locations based on 2D correspondences between images. Time-varying SfM problems have been studied in the context of ordered image-sequences of objects in motion (Ge and D'Zmura, 2003), while we work with an unordered (both spatially and temporally) collection of images. Although reasoning about visibility and occlusions has previously been applied to view synthesis from multiple images (Jelinek and Taylor, 2002), surface reconstruction (Taylor, 2003), and model-based self-occlusion for tracking (Sigal and Black, 2006), it has not been used in the context of temporal sorting.

The earliest work on temporal reasoning involved the development of an interval algebra describing the possible relationships between intervals of time (Allen, 1983). A number of specific temporal reasoning schemes were later captured by temporal constraint networks (Dechter et al., 1991) which pose the temporal inference problem as a general constraint satisfaction problem. Such networks are often used for task scheduling, given constraints on the duration and ordering of the tasks. Efficient solutions to temporal constraint networks rely on sparsity in the network, whereas our problem amounts to handling

a fully connected network. Uncertainty was later introduced into temporal constraint networks (Dubois et al., 2003, 1996; Badaloni et al., 2004) by relaxing the requirement that all constraints be fully satisfied.

4.3 Overview of Approach

We are interested in inferring the temporal ordering of images as one step in a system for producing time-varying 3D models of cities from historical photographs. As summarized in Figure 31 the process begins by performing feature detection and matching on a set of input photographs, followed by SfM to recover 3D points and camera poses. The feature detection and SfM steps are beyond the scope of this chapter and we do not discuss them in detail here, other than to say that in this work the feature detection and matching are performed manually (see Chapter 2 for more details).

In this chapter, we focus on the problems of visibility reasoning, temporal ordering, and time-varying 3D model construction as highlighted in Figure 31. Our method takes 3D points and camera poses as input and uses them to compute a matrix describing the visibility of each 3D point in each image (Section 4.4). The temporal ordering of the images is then recovered by reordering the columns of this visibility matrix in a CSP framework (Section 4.5). Finally, the inferred temporal ordering is used to visualize a 4D model (space + time) of the changing city (Section 4.6).

4.4 Visibility Reasoning

The problem we will address is inferring the temporal ordering of a set of n unordered images $I_{1..n}$ registered to a set of m 3D points $X_{1..m}$. The key to inferring temporal order from a collection of historical urban images is that different sets of 3D structures exist in the same place in the world at different points in time. Thus, we must determine which structures exist in each image, and to do this we must reason about the visibility of each 3D point in each image. We show here how to encode the information provided by each image

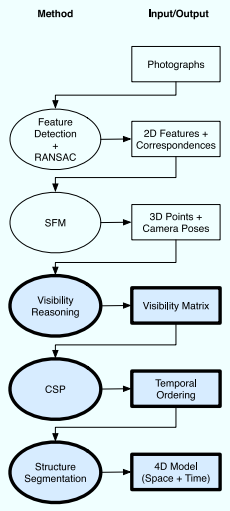


Figure 31: Overview of Approach. A fully automated system for building a 4D model (3D + time) of a city from historical photographs would consist of all these steps. Here, we concentrate on the highlighted steps of visibility reasoning and constraint satisfaction to infer a temporal ordering of images which can then be used to construct the 4D model.

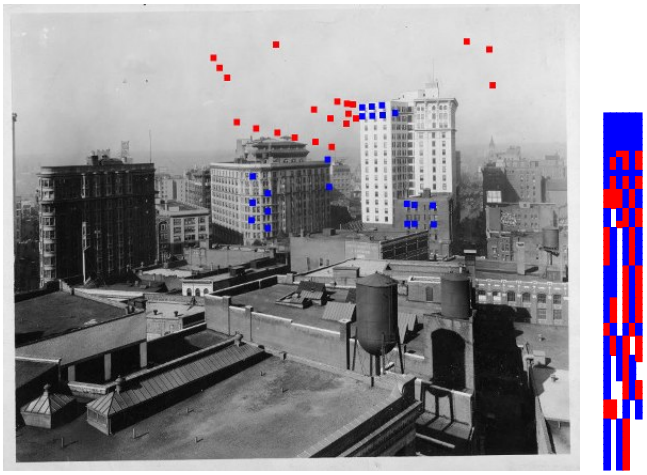


Figure 32: Point Classification. In each image, every 3D point is classified as *observed* (blue), *missing* (red), *out of view* (white) or *occluded* (white). The missing points belong to buildings that do not yet exist at the time the photograph was taken. Classifications across all images are assembled into a visibility matrix (right) which is used to infer temporal ordering. Each column of the visibility matrix represents a different image, while each row represents the visibility of a single 3D point across all images.

I_j about every 3D point X_i in a visibility matrix.

4.4.1 Visibility Classification

To determine whether a building exists at the time an image was taken, we reason about the visibility of each 3D point on that building. Assuming known projection matrices $P_{1..n}$ for each of the n cameras $C_{1..n}$ corresponding to images $I_{1..n}$, every 3D point can be classified in each image as *observed*, *missing*, *out of view*, or *occluded* as follows. If a measurement u_{ij} exists for point X_i in image I_j, the point is *observed*. If the projection $x_{ij} = P_j \begin{bmatrix} X_i \\ 1 \end{bmatrix}$ of point X_i in image I_j falls outside the field of view of the camera (as defined by the width and height of the corresponding image), the corresponding point is classified as *out of view* for that image. If the projection x_{ij} is within the field of view of the camera but no measurement u_{ij} exists, the point may be classified either as *missing* or *occluded*, and further work is required to determine which classification is correct (see Section 4.4.2).

The intuition behind this classification is that we want to know whether the physical structure corresponding to point X_i existed at the time that image $I_j$was captured. If it does not appear where we expect it to be, either it did not exist at the time (*missing*) or else something is blocking our view of it (*occluded*). We discuss how to distinguish between these two cases in the next section.

4.4.2 Occlusion

We can also use occlusion reasoning to determine why a building might not appear in a given image. To this end, we assume that the 3D points $X_{1..m}$ correspond to a sparse sampling of the surface of a number of solid structures in the scene. For every triplet of points, the triangle $X_aX_bX_c$ that they define may or may not lie along the surface of a solid structure. If we can find a triangulation of these points that approximates the solid structure, the faces of such a mesh will occlude the same set of points occluded by the physical structure, and these occluding faces can be used to distinguish between points that

are *missing* and *out of view*.

Inspired by (Faugeras et al., 1990) and the image-consistent triangulation method of (Morris and Kanade, 2000), we proceed as follows: For each image I_j, we compute the Delaunay triangulation of the measurements u_{ij} in that image. Each 3D triangle corresponding to a face in the Delaunay triangulation is a potential occluder and for each triangle, we test whether it fails to occlude any *observed* points in the scene. That is, if a face is intersected by a line segment O_jX_i from any camera's center of projection O_j to any observed 3D point X_i corresponding to a measurement u_{ij}, it is removed from the potential pool of occluders. The intuition behind this approach is that if the triangle was a true occluder, it would have blocked such a measurement from being observed. After testing all faces against all observed points X_i in all images I_j, we are left with a subset of triangles which have never failed to block any 3D point from view, and we treat these as our occluders.

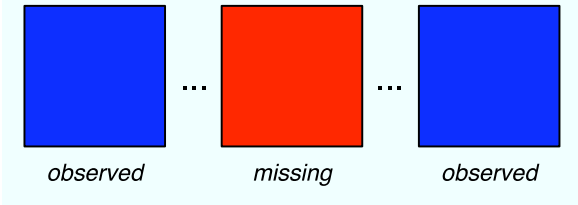
To determine whether a point X_i is *missing* or *occluded* in a given image I_j, we construct a line segment from the center of projection O_j of camera C_j to the 3D point X_i. If this line segment O_jX_i intersects any of the occluding triangles, the point is classified as *occluded*. Otherwise the point is classified as *missing*, indicating that the point X_i did not exist at the time image $I_j$was captured.

4.4.3 Visibility Matrix

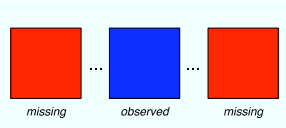
Finally, we can capture all this information in a convenient data structure—the visibility matrix. We construct an $m \times n$ visibility matrix V indicating the visibility of point X_i in image I_j as

$$v_{ij} = \begin{cases} +1 & if\, X_i\, is\, observed\, in\, I_j \\ -1 & if\, X_i\, is\, missing\, in\, I_j \\ 0 & if\, X_i\, is\, out\, of\, view\, or\, occluded\, in\, I_j \end{cases}$$

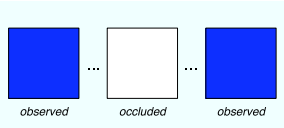
See Figure 32 for an example of such a visibility matrix. In all figures, the value $+1$ is indicated with a blue dot, -1 with a red dot, and 0 with a white dot. Note that the columns of such a matrix correspond to entire images, while the rows correspond to single 3D points.



(a) violates constraints



(b) satisfies constraints



(c) satisfies constraints

Figure 33: Visibility constraints. The columns of the visibility matrix must be reordered such that the situation in (a) never occurs – it should never be the case that some structure is visible, then vanishes, then appears again. Rather, we expect that buildings are constructed and exist for some amount of time before being demolished as in (b). Note that the constraint in (a) does not rule out the situation in (c) where structure becomes occluded.

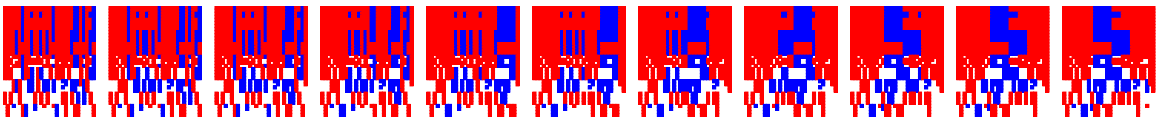


Figure 34: Local Search starts from a random ordering and swaps columns and groups of columns in order to incrementally decrease the number of constraints violated. Here, 30 images are ordered by taking only 10 local steps.

4.5 Constraint Satisfaction Problem

We pose the temporal ordering problem as a constraint satisfaction problem (CSP), where constraints are applied to the visibility matrix of the given scene. Specifically, once a visibility matrix V is constructed, the temporal ordering task is transformed into the problem of rearranging the columns of V such that the visibility pattern of each point is consistent with our knowledge about how buildings are constructed. Our model assumes that every point X_i is associated with a building in the physical world, and that buildings are built at some point in time T_A, exist for a finite amount of time, and may be demolished at time T_B to make way for other buildings. We also assume that buildings are never demolished and then replaced with an identical structure. These assumptions gives rise to constraints on the patterns of values permitted on each row in V.

The constraints on the visibility matrix can be formalized as follows: on any given row of V, a value of -1 may not occur between any two $+1$ values. This corresponds to the expectation that we will never see a building appear, then disappear, then reappear again, unless due to occlusion or being outside the field of view of the camera (see Figure 33). The valid image orderings are then all those that do not violate this single constraint.

Because we have expressed the temporal ordering problem in terms of constraints on the visibility matrix, we can use the general machinery of CSPs to find a solution. A common approach to CSPs is to use a recursive backtracking procedure which explores solutions in a depth first search order by assigning an image I_j to position 1, then another image to position 2, etc. At each step, the partial solution is checked and if any constraints are violated, the current branch of search is pruned and the method "backtracks" up one level to continue the search, having just eliminated a large chunk of the search space. Given that our problem has $n!$ solutions (i.e., factorial in the number of images n), this method becomes computationally intractable for even relatively small numbers of images.

4.5.1 Local Search

CSPs can also be solved using a local search method to get closer and closer to the solution by starting at a random configuration and making small moves, always reducing the number of constraints violated along the way. This solution has been famously applied to solve the n-queens problem for 3 million queens in less than 60 seconds (Sosic and Gu, 1991).

For our problem, a local search is initialized with a random ordering of the images, corresponding to a random ordering of the columns in the visibility matrix V. At each step of the search, all local moves are evaluated. In our case, these local moves amount to swapping the position of two images or of two groups of images by rearranging the columns of the matrix V accordingly. In practice, swapping larger groups of images allows solutions to be found more quickly, preserving the progress of the search by keeping constraint-satisfying sub-sequences of images together.

During local search, we consider a number of candidate orderings of the columns of the visibility matrix, where different arrangements of columns will violate different numbers of constraints. As described above, a constraint is violated if, on a given row, a point is classified as *missing* between two columns in which it was *observed*. The best local move is then the move that results in the ordering that violates the fewest constraints of all the candidate local moves being considered. If there is no move which decreases the number of constraints violated, we reinitialize the search with a random ordering and iterate until a solution is found. Once an ordering of the columns is found that violates no constraints, the temporal ordering of the images is exactly the ordering of the columns of the visibility matrix. Figure 34 demonstrates the progress of such a local search.

4.5.2 Properties of Ordering Solutions

Solving the above constraint satisfaction problem may give us more than just one possible temporal ordering of the images. For the n images, there may be r eras in which different combinations of structures coexist. If $r < n$, there is more than one solution to the constraint satisfaction problem. In particular, any two images captured during the same era may be swapped in the ordering without inducing any constraint violations in the visibility matrix.

In addition, there is a second class of solutions for which time is reversed. This is because any ordering of the columns that satisfies all constraints will still satisfy all constraints if the order of the columns is reversed. In practice, one can ensure that time flows in the same direction for all solutions by arbitrarily specifying an image that should always appear in the first half of the ordering. This is analogous to the common technique of fixing a camera at the origin during structure from motion estimation.

4.5.3 Dealing with Uncertainty

The above formulation depends upon an explicit decision as to the visibility status of each point in each image, and cannot deal with misclassified points in the visibility matrix. For example, if a point is not observed in an image, it is crucial that the point receives the

correct label indicating whether the point no longer existed at the time the image was taken, or whether it was simply occluded by another building. If a single point is misclassified in one image, it may cause all possible orderings to violate at least one constraint, and the search will never return a result.

The ideal case, in which there are no occlusions and no points are out of view, will rarely occur in practice and there are a number of ways a point might be misclassified:

- Points that really should have been *observed* might, due to failure at any point during automated feature detection or due to missing or damaged regions of historical images, be classified as *missing*.
- Points that were *occluded* by un-modeled objects (such as trees or fog) may falsely be labeled *missing*.
- Points that were really *occluded* may fail to be blocked by occlusion geometry due to errors in SfM estimation, and instead be falsely labeled as *missing*.
- Points that are truly *missing* may be falsely explained away as *occluded*.

In practice, some combination of all these errors may occur.

We achieve robustness to misclassified points without introducing any additional machinery. CSPs can implicitly cope with this kind of uncertainty by relaxing the requirement that all constraints be satisfied. We modify the local search algorithm to return the ordering that satisfies more constraints than any other after a fixed amount of searching. Under such an approach, we can no longer be absolutely certain that the returned solution is valid, but we gain the ability to apply the approach to real-world situations.

4.5.4 Structure Segmentation

In order to build a convincing 3D model of a city, we need to segment the 3D point cloud that results from SfM into a set of solid structures. In fact, we can use the visibility matrix V to extract such a building segmentation directly from the recovered image ordering. Once

the columns have been reordered using local search, similar visibility patterns become apparent across the rows of the matrix. This is due to the fact that multiple 3D points originate from the same physical structures in the world, and thus come in and out of existence at the same time. This is made more apparent by reordering the rows of the visibility matrix to group points that share times of appearance T_A and disappearance T_B. Such a reordering amounts to segmenting the 3D point cloud into disjoint sets. By taking the 3D convex hull of each cluster of points, we get approximate scene geometry which can be textured and used for further synthesis of new views in space and time (see Figure 35).

4.6 Results

We tested our method on a set of images of Atlanta collected over the period from 1897 to 2006. For the results presented here, feature detection and matching were performed manually. Given a set of 2D correspondences across images, the remaining steps of the algorithm beginning with SfM (see Figure 31) are performed automatically.

In our first experiment, we find a temporal ordering for 6 images of a scene containing 56 3D points (Figure 36). In this case, we purposely chose photographs with clear views of all structure points, meaning that none of the points are misclassified in the visibility matrix and an exact solution to the ordering is guaranteed. Due to the small number of images, we

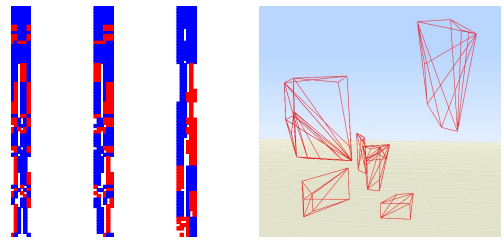


Figure 35: Structure Segmentation. Beginning from a random ordering of the visibility matrix (a), local search re-orders the columns to the correct temporal ordering (b), and then rows are re-ordered to group 3D points that appear and disappear at the same times (c). We compute 3D convex hulls of each group of points to get solid geometrical representations of buildings in the scene (d).

Figure 36: Inferred temporal ordering of 6 images. In the case where there are no occlusions of observed points, we can guarantee that a solution exists that violates no constraints. The ordering shown is one of 24 orderings that satisfy all constraints. The other solutions involve swapping sets of images that depict the same set of structures and reversing the direction of time.

Figure 37: Inferred ordering of 20 images. Despite many misclassified points, the presence of un-modeled occlusions such as trees, and a solution space factorial in the number of images ($20! \approx 2.4 \times 10^{18}$), an ordering consistent with the sets of visible buildings is found by using local search to find the ordering that violates the fewest constraints. In such a case, there is no single solution which satisfies all constraints simultaneously.

perform an exhaustive back-tracking search to find all possible ordering solutions. Back-tracking search finds that out of the $6! = 720$ possible orderings, there are 24 orderings which satisfy all constraints, one of which is shown in Figure 36. The 24 solutions are all small variations of the same ordering—images 1 and 2 may be interchanged, as may images 4, 5, and 6, and finally the entire sequence may be reversed such that time runs backwards. For this small problem, the search takes less than one second.

In our second experiment, we deal with a more difficult group of 20 images of a scene consisting of 92 3D points (Figure 37). These images contain a number of misclassified points due to occlusions by trees and un-modeled buildings, as well as errors in the estimation of 3D point locations and camera positions by SfM. As such, we do not expect to find an ordering that satisfies all constraints, so we instead use 1000 iterations of local search to find the ordering which violates the fewest constraints. For each iteration of local search, we begin from a new random ordering of the images. Note the number of iterations of search (1000) is considerably smaller than the number of possible orderings, in this case $20! \approx 2.4 \times 10^{18}$. This local search returns an ordering (Figure 37) for which constraints are violated on 15 of the 92 rows of the visibility matrix. In the absence of any ground truth

dates for the images, and with no exact solution to the CSP in this case, it is difficult to evaluate the quality of the returned ordering. However, despite the large number of constraints violated, the ordering returned is consistent both with the sets of buildings which appear in each image and with the known dates of construction and demolition for all modeled buildings in the scene. The ordered visibility matrix for this experiment is shown in Figure 38.

In our third experiment, to simulate a larger problem, we synthesize a scene containing 484 randomly distributed 3D points and 30 cameras placed in a circle around the points. Each point is assigned a random date of appearance and disappearance, while each camera is assigned a single random date at which it captures an image of the scene. The resulting synthetic images only show the 3D points that existed on the date assigned to the corresponding camera. The size of the solution space ($30! = 2.65 \times 10^{32}$) necessitates local search for this problem. Starting from a random ordering, a solution that violates no constraints is found just 26 local moves away from the random initialization, taking less than one minute of computation. In contrast to the previous experiment, a solution is quickly found for this synthetic scene (without the need to reinitialize the search) because no points are misclassified for the synthesized images.

Finally, we use the structure segmentation technique described in Section 4.5.4 to automatically create a time-varying 3D model from the 6 images in Figure 36. After ordering the columns of the visibility matrix to determine temporal order, we reorder the rows to group points with the same dates of appearance and disappearance. We then compute the convex hulls of these points and automatically texture the resulting geometry to visualize the entire scene (see Figure 39). Textures are computed by projecting the triangles of the geometry into each of the 6 images and warping the corresponding image regions back onto the 3D geometry.

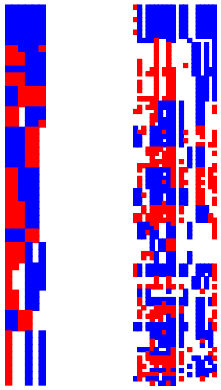


Figure 38: Ordered visibility matrices for sets of 6 images (left) and 20 images (right). The ordering of the 6 images on the left was found with backtracking search and satisfies all constraints. The ordering of the 20 images on the right violates the fewest constraints of all solutions found with 1000 iterations of local search. In the latter case, misclassified points caused by un-modeled occlusions lead to a situation in which no ordering can simultaneously satisfy all constraints.

4.7 Discussion

The computation time required for local search depends upon several factors. The main computational cost is computing the number of constraints violated by a given ordering of the visibility matrix, which increases linearly with m the number of points in the scene and n the number of images being ordered. In addition, at each step of local search, the number of tested orderings increases with n^2 since there are $\frac{(n)(n-1)}{2}$ ways to select two images to be swapped.

As demonstrated in the above experiments, the amount of computation also varies inversely with the number of valid orderings for a given visibility matrix. For ordering problems that admit many solutions, the random initialization of local search will often be close to some valid ordering, and will thus solve the problem quickly. This is, in fact, the key to the success of local search on the n-queens problem of (Sosic and Gu, 1991), where the number of solutions actually increases with the size of the board. However, when there are very few solutions (or no exact solution, as in the above 20-image experiment), local search may require a large number of iterations until a random ordering is chosen that can reach the true solution using only local moves.

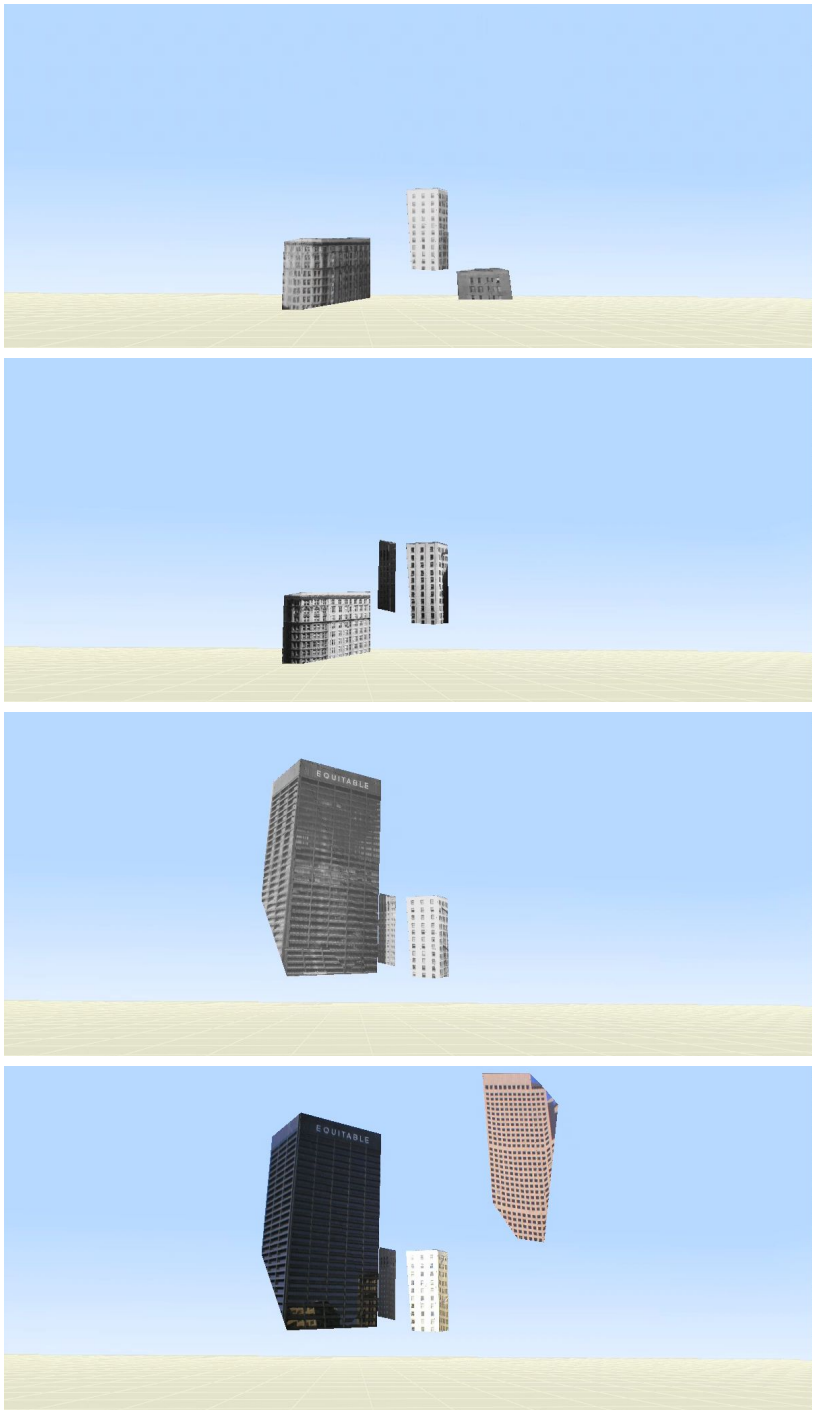


Figure 39: Time-varying 3D model. Here, we see the scene as it appeared at 4 different times from the same viewpoint. This result is generated automatically given 2D point correspondences across 6 unordered images as input. We perform SfM, determine occluding surfaces, compute the visibility matrix, solve the CSP using local search to infer temporal ordering, group points based on common dates of existence, compute 3D convex hulls, and texture triangles based on where they project into each image.

Finally, note that the nature of the dates we infer for scene structure is abstract. For example, consider the building depicted in the first image in Figure 37. Rather than inferring that this building existed from 1902 to 1966, we can only infer that it existed from the time of Image 1 to the time of Image 13 (where images are numbered by their position in the inferred temporal ordering). Without additional knowledge, this is the most we can confidently say about when the building existed. When a human inspects a historical photograph, he or she may assign a time to it by identifying objects in the scene with known dates of existence—this may include known buildings, but also more abstract concepts such as the style of automobiles, signs, or the clothing of people depicted in the image. This suggests that a machine learning approach may be required if we hope to assign estimates of absolute dates to each image.

4.8 Conclusion

In this chapter, we have shown that constraint satisfaction problems provide a powerful framework in which to solve temporal ordering problems in computer vision, and we have presented the first known method for solving this ordering problem. The visibility reasoning approach introduced here forms the basis of the temporal inference methods used throughout this dissertation. In the next chapter, we begin to explore methods of incorporating absolute dates into the temporal inference process in order to move beyond simple ordering of images.

Chapter V

INCORPORATING IMAGE DATES: ABSOLUTE TEMPORAL INFERENCE

In the previous chapter, we showed how visibility information about a reconstructed 3D scene can be exploited to recover a temporal ordering of images of the scene. Now, we will discuss methods of incorporating absolute dates into the temporal inference problem. The methods introduced here rely on a complete and accurate set of observations of a set of buildings in each image, and thus we use manual point correspondences and interactively constructed 3D building models.

5.1 Problem

Large collections of archival photographs are going online at an increasing rate. There has been much recent interest in exploiting large online image collections for 3D reconstruction (Snavely et al., 2006), scene completion (Hays and Efros, 2007), and geographic localization (Hays and Efros, 2008). Most of this work has concentrated on user-submitted collections of present-day photographs uploaded by users over the past several years. As older institutions digitize their archival photo collections, millions of photographs from the late 19th and 20th centuries are becoming available online. While these images present a number of interesting vision challenges, we focus here on one task in particular: determining the date on which a photograph was captured, a task currently performed by human experts.

We approach the problem of automatically dating urban photographs via both global appearance and 3D structure. Our appearance-based approach is motivated by the observation that images from the 1800s simply look different from modern images (see Figure 40),

Figure 40: To estimate the dates of urban photographs, we reason both about structural changes over time and changes in the appearance of cities throughout many decades. Here we see the same city from similar viewpoints in 1864, 1906, 1973, and 2003.

and we would like to exploit this. On the other hand, we know from the previous chapter that 3D reconstructions induce temporal ordering constraints between images. Thus, we would also like to transfer temporal information from dated to undated images via commonly observed 3D structures. We adopt both of these approaches and tie them together in a continuous optimization framework.

5.2 Related Work

Dating of historical photographs is a task currently performed by human experts at the cost of much effort. There are several broad categories of techniques currently used to date photographs, including both physical and visual examination. Physical examination techniques include examination of the photograph for specific chemicals (e.g. optical brightening agents), paper fiber characteristics, paper size, and manufacturer logos (Messier, 2005). On the other hand, visual examination approaches rely on identifying the clothing, hairstyles, and accessories of individuals as belonging to a specific era in history (Pols, 2002). In fact, books full of examples of historical photographs with known dates (Moorshead, 2000) are a common tool used in the dating of old photographs. The appearance-based dating methods we use here are inspired both by these photograph dating tools and by recent work on example-based location recognition (Hays and Efros, 2008). The image dating task can also be seen as a variant of automatic image annotation (Pinar Duygulu and Forsyth, 2002)

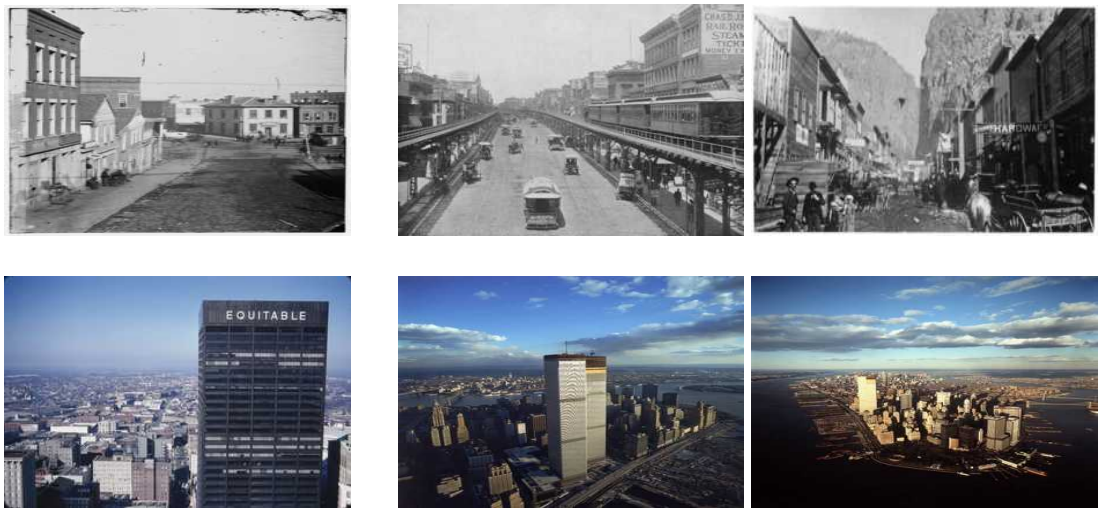


Figure 41: Appearance-Based Matching. For each test image (left), the best two matches from the LIFE database are shown at right. Matching is performed on texton histograms computed for each image, with a texton vocabulary size of 100. The key idea is that images which were taken around the same date (on a historical time scale) should be similar in appearance.

specific to the temporal domain.

The 3D structure-based dating techniques we adopt here are built upon the temporal ordering techniques introduced in (Schindler et al., 2007) and discussed in the previous chapter. While this relative temporal inference method focuses on a purely abstract ordering of images, without respect to absolute dates of any kind, here we are interested in inferring a specific date in history for a given photograph. Accordingly, while relative temporal inference uses a discrete optimization technique to solve the temporal ordering problem, we introduce a continuous optimization framework to perform inference on continuous dates.

5.3 Appearance Matching

There are several reasons to believe that appearance matching is a feasible approach to dating images. First is that when presented with historical images (even of unfamiliar locations), humans are often able to estimate the era from which the photograph originates. Second, similar techniques have shown to perform many times better than random chance on the seemingly more difficult task of global location recognition (Hays and Efros, 2008)

and object recognition (Torralba et al., 2008). In addition to visual changes in architectural styles, building materials, storefronts, and vehicles, things like typical viewpoints have changed over the decades with the arrival of tall buildings and aerial photography, all of which lead to changes in the statistics of images captured at different points in history.

We begin by building a database of date-labeled images. To do so, we use the LIFE magazine photo archive recently made available by Google, which consists of the magazine's photo archive from the 1840s to the present. Since not all images in the database are of urban locations, we perform an automated search for images with the descriptive tags "street scenes", "buildings", or "cities" and a decade tag from "1840s" to "2000s". The result is a set of 795 images of cities, each of which is annotated with a specific historical date. Note that we exclude from this database all images of the city from which our testing set of images were captured. This step was taken to ensure that the features upon which we are matching are location-independent.

We perform appearance matching using a texton (Renninger and Malik, 2004) vocabulary which has shown good performance for the location recognition task (Hays and Efros, 2008). We learn a vocabulary of 100 textons via k-means clustering on responses to a bank of filters (at 6 orientations and 3 scales). Each image is represented by a normalized histogram of texton frequencies.

For a given test image, we find its K nearest neighbors (by chi-squared distance) in the database of date-labeled images. We are interested not just in estimating a single date for each image, but a distribution which encodes the uncertainty about the date of a given image. We estimate this density with the mean and variance of the K nearest neighbors. Thus while some images will match mostly 1880s images, others will match a broad range, meaning they could have been taken any time from 1950 to today, for example. We will make use of this variance information in our continuous optimization framework (Section 5.5).

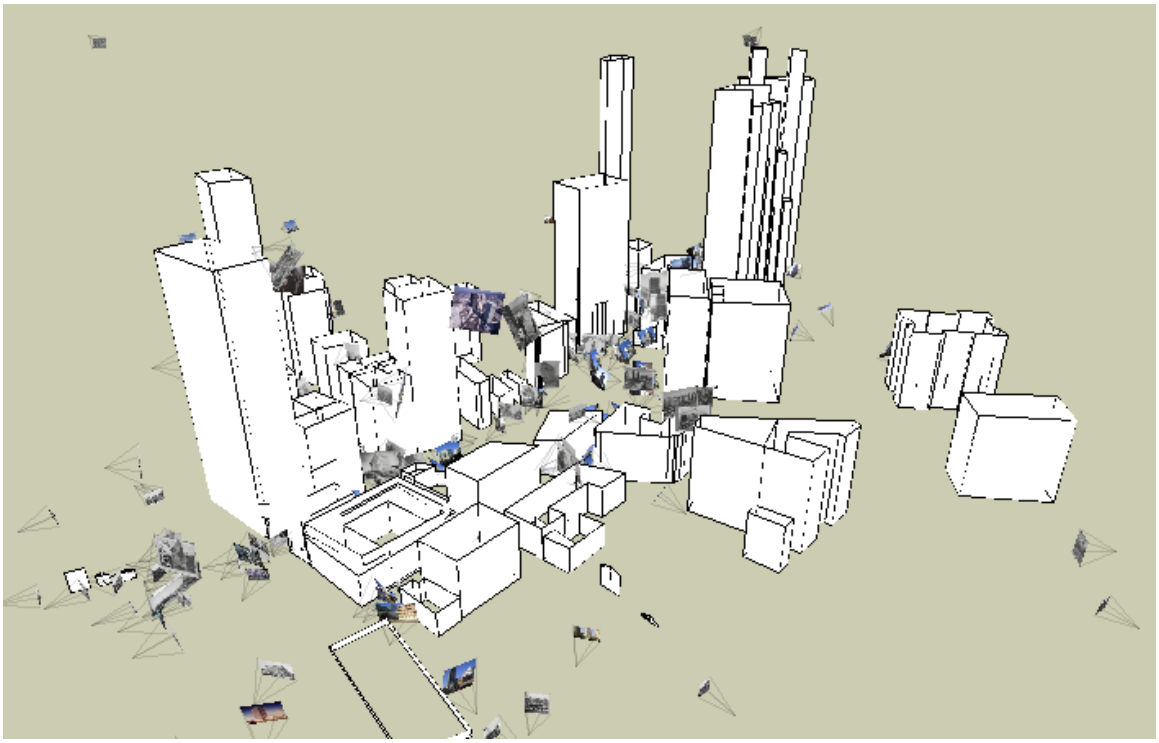


Figure 42: The structural visibility dating method relies on a set of images with associated geometry reconstructed from the images using structure from motion. As a part of the dating process, a date interval must be estimated for each structure. Note that not all of the structures pictured here existed simultaneously in history.

Note on Date Representation Given a photograph I_j labeled with a year y_j, month m_j, and day d_j, the date of the photograph $t_j \in \mathbb{R}$ is represented as $t_j = y_j + f(m_j, d_j)/365$. This is the value of the year plus the fractional amount of a year accounted for by the day and month where $f()$ is a function from month and day to sequential day of the year. We make this explicit because historical photographs are often labeled with a year only, for example 1917, in which case we only know that the true date of the photograph lies within an interval $t_j \in [1917.0, 1918.0)$. In such a case, we take the midpoint of the interval as the value of t_j for the sake of convenience.

5.4 Structural Visibility Dating

We introduce a *structural visibility dating* method which uses a 3D reconstruction of a scene to transfer dates between images involved in the reconstruction. As a preliminary

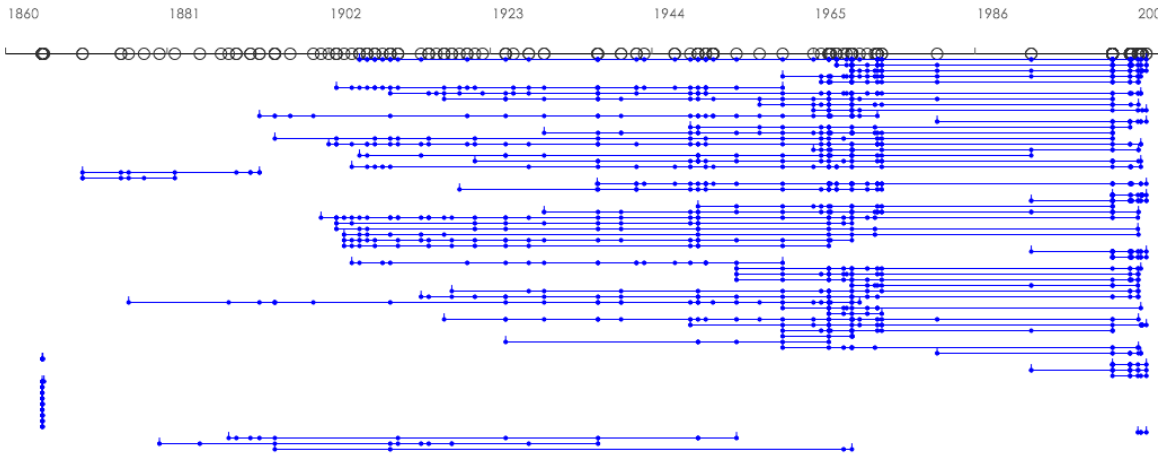


Figure 43: Structure Intervals constructed from date-labeled images. Given known dates for a subset of the images, the correspondences used to perform structure from motion are then used to put bounds on the date intervals that describe when each 3D structure existed. Each blue bar in the above image represents the date interval for one of the 3D structures in Figure 42. Black circles indicate dates of images, while blue dots indicate that a specific structure was observed in specific image on a specific date.

step, we use a set of historical photographs to construct a sparse 3D city model (see Figure 42) via structure from motion with a set of human-provided point correspondences. A human also interactively connects 3D points that belong to the same building so that a rough polygonal model of each building is constructed (as described in Chapter 2) both for occlusion modeling purposes and to tie together observations of different points on the same building. We consider each building to be an object O_i, with m objects in total.

Building on the relative temporal inference concepts introduced in the previous chapter, we compute a visibility matrix V_{ij} which indicates the observation status of each object O_i in each image I_j (see Figure 44). Observations may be positive (the object was observed in the image), negative (the object was provably absent, based on occlusion reasoning), or there may be no observation at all for a given ij pair. We use the same occlusion reasoning procedure as in the previous chapter, except that the interactively constructed building geometry now acts as our occlusion geometry as well.

We now have a set of images I, a set of buildings O, and an indication of which buildings were observed in each image V. Now, we would like to determine the absolute date t_j of

one of the images I_j. If we have no knowledge of the dates of any photographs or of the construction/demolition dates of the buildings, then we are stuck. However, if just a few of the image dates t_j are known, then we can infer dates of the unlabeled images via a bootstrap method that first estimates *object date intervals* (a_i, b_i) from labeled images, and then estimates *image dates* t_j from the buildings observed in each image.

There are four important dates in the history of any object O_i: the earliest and latest positive observations (p_{i1} and p_{i2}), and the latest and earliest negative observations ($n_{i1}$and n_{i2}) preceding and following these positive observations, respectively. Given these four dates, the beginning date a_i of object O_i lies within the range (n_{i1}, p_{i1}) while the end date b_i lies within the range (p_{i2}, n_{i2}). Even in the absence of negative observations n_{i1} and n_{i2} (negative observations may be rare), we can still say that object O_i definitely existed from time p_{i1} to p_{i2}, and that the date of any photograph observing such a point may lie within the same range. Thus, we compute the values p_{i1}, p_{i2}, n_{i1}, and n_{i2} for all objects $O_{i=1:m}$ across all dated images $I_{j=1:n}$, and for the time being, we assume that each object's beginning date is $a_i = p_{i1}$ and that each object's ending date is $b_i = p_{i2}$.

Given these date intervals (a_i, b_i) for each 3D object, the key idea is that when more than one structure is observed in a given undated image, the valid range of dates for the photograph is given by *intersecting* the intervals of existence of all observed buildings. We can then, for example, take the midpoint of this interval as an estimate of the image date. See Figure 43 for an example of building date intervals constructed from a set of partially dated images.

This *structural visibility dating* method is quite simple, and amounts to asking the question: "I see five buildings in this image. When did they all exist together?" Note that this method produces the minimal photograph date range estimates that are consistent for the observed structures. However, since the building intervals themselves are based on the dates of database images in which the buildings are observed, this method may still fail if a test image is presented which predates the earliest database image of a given building,

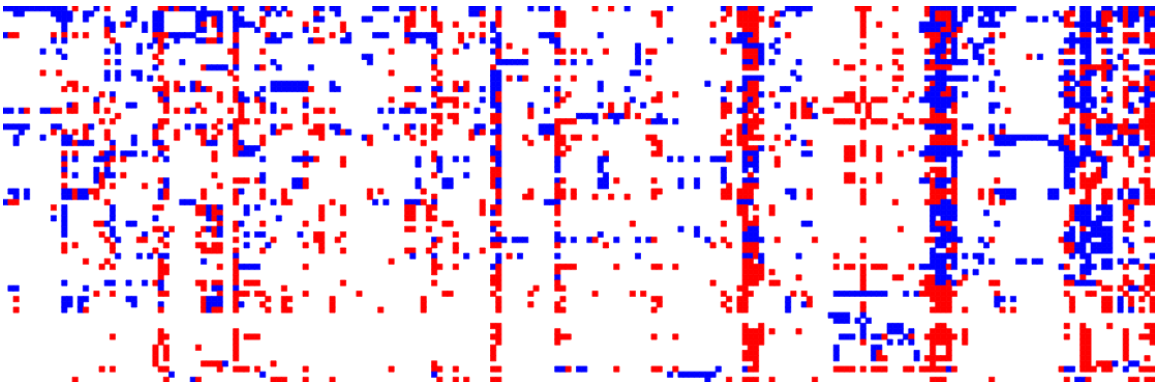


Figure 44: Visibility Matrix. The 70x212 visibility matrix for a collection of 212 images of a city with 70 buildings. Blue dots indicate positive information while red dots indicate negative information. Columns correspond to images, while rows correspond to 3D structures, in this case buildings. Matrix sparsity is 81.7%.

for example. More specifically, if the true date of a test image falls in the range (n_{i1}, p_{i1}) or (p_{i2}, n_{i2}) for any of the observed objects, then the estimated date interval for the image will be too small to contain the true date of the image because of our choice to base a_i and b_i on positive observations alone. On the other hand, the estimated image date intervals resulting from this technique can be quite broad. For example, if the two buildings visible in an image have co-existed for 80 years, then our estimate for the photograph's date has the same range. This is an argument in favor of using appearance-based methods to push our estimate to a specific date within that 80 year span. In the next section, we introduce a method for combining structural and appearance information in a continuous optimization framework.

5.5 Continuous Optimization

We introduce a *continuous optimization* method for temporal inference on images and 3D structures. Specifically, we are interested in inferring a continuous time for each image and a continuous time interval for each 3D structure. Continuous optimization allows us to estimate these time values directly, while at the same time avoiding combinatorial explosion in the search space of discrete methods such as the stochastic greedy local search used in our relative temporal inference method.

It is a useful analogy to understand this continuous optimization method as a form of temporal bundle adjustment (Triggs et al., 1999) in that we are simultaneously seeking a date for each camera and a date interval for each 3D structural element which minimizes an error induced between the model and our observations, starting from some initial estimate of the model parameters. Whereas the structural visibility dating method (Section 5.4) estimated building date intervals from photographs with *known* dates, the continuous optimization method takes into account information from undated photographs as well. This property allows continuous optimization to theoretically succeed in the presence of very little information: fixing a temporal origin and scale is all that is required (i.e. two dated photographs), although in practice more information leads to better date estimates.

This approach allows us to take into account a number of types of information:

- Dates from dated photographs
- Structure observations (both positive and negative) from dated *and* undated photographs
- Appearance-based date estimates for undated photographs

thus allowing us to combine all the methods outlined above.

As laid out in Chapter 1, given a set of n images $I_{1..n}$ registered to a set of m 3D objects $O_{1..m}$, we wish to estimate a time t_j associated with each image, and a time interval (a_i, b_i) associated with each 3D structure. We use the variable T to represent the set of all the individual times t_j and time intervals (a_i, b_i) for every image and 3D structure. We choose to minimize the squared error between the observations in the visibility matrix V and the expected value of these observations given the current model parameters T.

$$\min_{T} \sum_{ij} [V_{ij} - E(V_{ij})]^2$$

Here, we are only concerned with those entries in the visibility matrix for which there is either positive or negative information about a given structure in a given image. If structure

is *out of view* or *occluded*, it does not affect our solution. We only want to ensure that all *observed* and *missing* observations agree with the temporal variables of the model. In accordance with this goal, and in a slight departure from the previous chapter, we give *observed* objects in the visibility matrix a value of 1 and *missing* objects a value of 0. Therefore, we compute the expected value of the observations as:

$$E(V_{ij}) = P(a_i \leq t_j < b_i) * 1 + [1 - P(a_i \leq t_j < b_i)] * 0$$

The expression $P(a_i \leq t_j < b_i)$ is the probability that an image I_j was captured while structure X_i existed. If there was no uncertainty on any of the estimated variables, then this probability would always evaluate to 1 or 0 according to the truth of the inequality $a_i \leq t_j < b_i$, and therefore the derivative of the error function would be everywhere zero or undefined. This is a problem if we hope to minimize this error function using iterative non-linear optimization methods. For mathematical convenience, we model the beginning and end dates of each structure as being distributed according to a logistic distribution, which is shaped like a heavy-tailed normal distribution. This makes our error function continuous and differentiable and allows us to use a non-linear optimization method like Levenberg-Marquardt to arrive at a solution for the temporal variables. Under this assumption, the probability $P(a_i \leq t_j < b_i)$ can be expressed as the product of two cumulative distribution functions on a_i and b_i, of the form $P(a_i \leq t_j)$ and $P(t_j < b_i) = 1 - P(b_i \leq t_j)$. Furthermore, the cumulative distribution function of a logistic distribution is expressible as the well-known logistic function, such that:

$$P(a_i \leq t_j < b_i) = \frac{1}{1 + e^{\frac{-(t_j - a_i)}{s}}} * \frac{1}{1 + e^{\frac{-(b_i - t_j)}{s}}}$$

In addition, we can incorporate information about the appearance of each image into the error minimization. As described above (Section 5.3), for the K nearest neighbors (in appearance) of a given image I_j, we compute the mean u_j and variance σ_j^2 of their associated dates. For the date t_j of each image, then, we have an additional error term:

$$\frac{(t_j - \mu_j)^2}{\sigma_j^2}$$

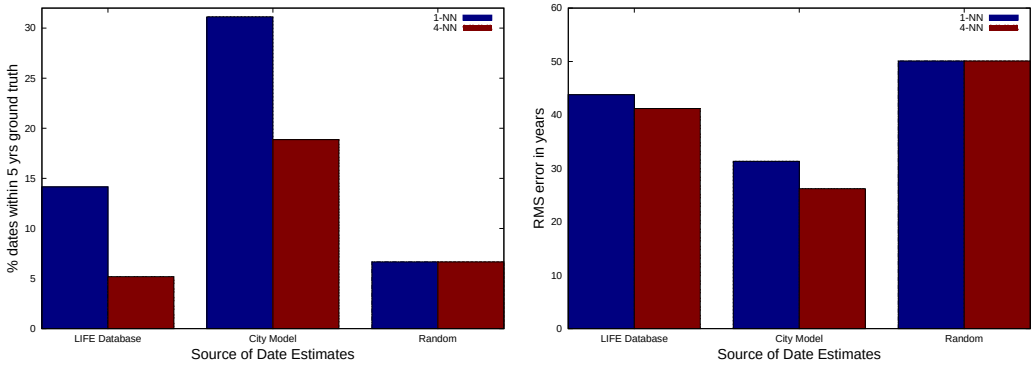


Figure 45: Appearance-Based Dating Performance. Performance is evaluated by the percentage of the 212 test images with estimated dates that fall within 5 years of ground truth. In all cases, taking the date of the single nearest neighbor maximized the number of estimates within 5 years of ground truth, while the mean of the four nearest neighbors minimized RMS error. The LIFE database consists of date-labeled images from around the world, while the City Model database uses images from the same city as the test images.

such that the final minimization is over the sum:

$$\min_{T} \sum_{ij} [V_{ij} - E(V_{ij})]^2 + \sum_{j} \frac{(t_j - \mu_j)^2}{\sigma_j^2}$$

Given an initial estimate of the temporal parameters T (for example from the structural visibility dating method outlined above), this formulation allows us find an optimal solution for all image times and structure time intervals which incorporates both structural and appearance-based constraints on the date of each image.

5.6 Results

We perform dating experiments on a test set of 212 images of Atlanta, spanning the years 1864 to 2008, (see Figures 40 and 41 for examples) for which we used the interactive 4D city modeling approach of Chapter 2 to recover camera poses and scene geometry (See Figure 42). The visibility matrix computed from this model is shown in Figure 44. Approximate dates of all 212 test images are known, either from EXIF tags on modern images, or from date annotations or educated guesses on historical images, and we treat these dates as ground truth. Using this data, we test all three methods outlined above, both individually and in combination.

Appearance-Based Dating performance is evaluated by the percentage of the 212 test images with estimated dates that fall within 5 years of ground truth. We also compute the root mean square (RMS) error to evaluate how far the date estimates are from ground truth on average. We tested appearance-based dating with two data sets: the LIFE database, which consists of date-labeled images from around the world, and what we call the City Model database, which involves matching each image from the test set against all other test set images in a leave-one-out fashion. The results of appearance-based dating are summarized in Figure 45, with specific examples shown in Figure 41.

We find that we can estimate the correct date within 5 years of ground truth in over 30% of cases when performing leave-one-out matching within the Atlanta database. This is a significant result as it indicates that without any specific image correspondences or 3D reconstruction involved, we can truly extract date information from texture and appearance alone for a number of images in our data set. Matching to the generic set of LIFE images is less successful, but still significantly better than random. In hindsight, it makes sense that matching within the city database would perform better than matching to a generic set of city images, since the texton histograms can encode appearance information about specific buildings in a city in addition to more generic texture information such as dirt road versus paved road. In addition, each city takes a unique path through its development, such that the temporal information in a photo may depend greatly upon its global location.

In all cases, taking the date of the single nearest neighbor maximized the number of estimates within 5 years of ground truth, while the mean of the four nearest neighbors minimized RMS error. We also note that the performance of our 1-NN texton matching is roughly on par with (Hays and Efros, 2008) which used a similar approach in the location domain, suggesting that improvements in image representation and matching will benefit both the dating and localization tasks equally. The results of (Hays and Efros, 2008) and (Torralba et al., 2008) also suggest that growing the size of our database is one route to better performance.

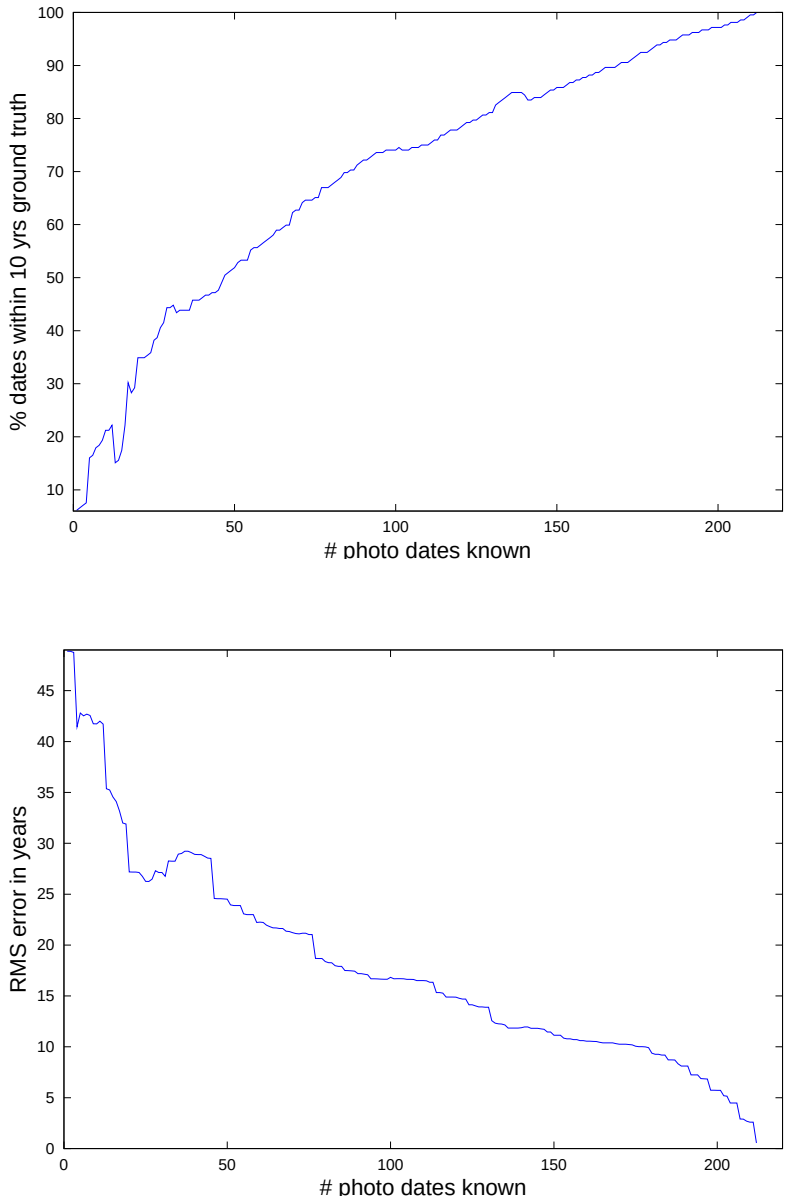


Figure 46: Structure Visibility Dating Performance. Dating performance improves as more labeled images are included in the model, shown here by the percentage of date estimates within 10 years of ground truth (top) and the root mean square error (in years) for date estimates (bottom). The structural visibility dating method is a two-step process which (1) estimates date intervals for structures (e.g. buildings) based on a limited number of date-labeled images which have observed these structures, and (2) estimates dates for unlabeled images based on the structures they observe.

Structural Visibility Dating performance is summarized in Figure 46. This method relies on the model reconstructed using SfM and assumes a certain number of images in the model have known dates. We vary the number of images with known dates from none to all images and show the effect on RMS error and percentage of date estimates within 10 years of ground truth in Figure 46. The benefit of structural visibility dating is clearly visible in the top graph of Figure 46 by the curve's height above the diagonal – for example, given known dates for 19% of the images (40 images), 49% of the images are dated within 10 years of ground truth. Though the 3D reconstruction of individual city models is clearly more labor intensive than appearance-based matching to a global database, these results indicate that the effort is well-rewarded in performance.

Continuous Optimization. We find that although our continuous temporal optimization method works well on synthetic scenes, it ultimately fails on real data due to the sparsity of observations in the visibility matrix. Though a disappointing result, this finding ultimately leads us to the successful methods of the next chapter, so all is not lost. We report experimental results here for the sake of completeness, and because the way in which this method fails is itself quite informative about the temporal inference problem.

Performance of continuous optimization is evaluated both on real and synthetic data. For a synthetic scene (see Figure 47) consisting of 30 images observing 20 buildings, we perturb the temporal parameters from their ground truth values, such that the RMS error on images dates is 18.8 years, with only 30% of images within 10 years of their ground truth value. We fix two of the images to their correct dates in order to provide a temporal origin and scale to the solution. We run 500 iterations of the Gauss-Newton algorithm to minimize the error between expected and observed visibility matrix terms as defined above. The result is a final RMS error (with respect to ground truth) of only 6.63 years with 80% of images within 10 years of their ground truth values. This is an excellent result which shows that our continuous optimization method is capable of solving temporal inference problems when given appropriate input.

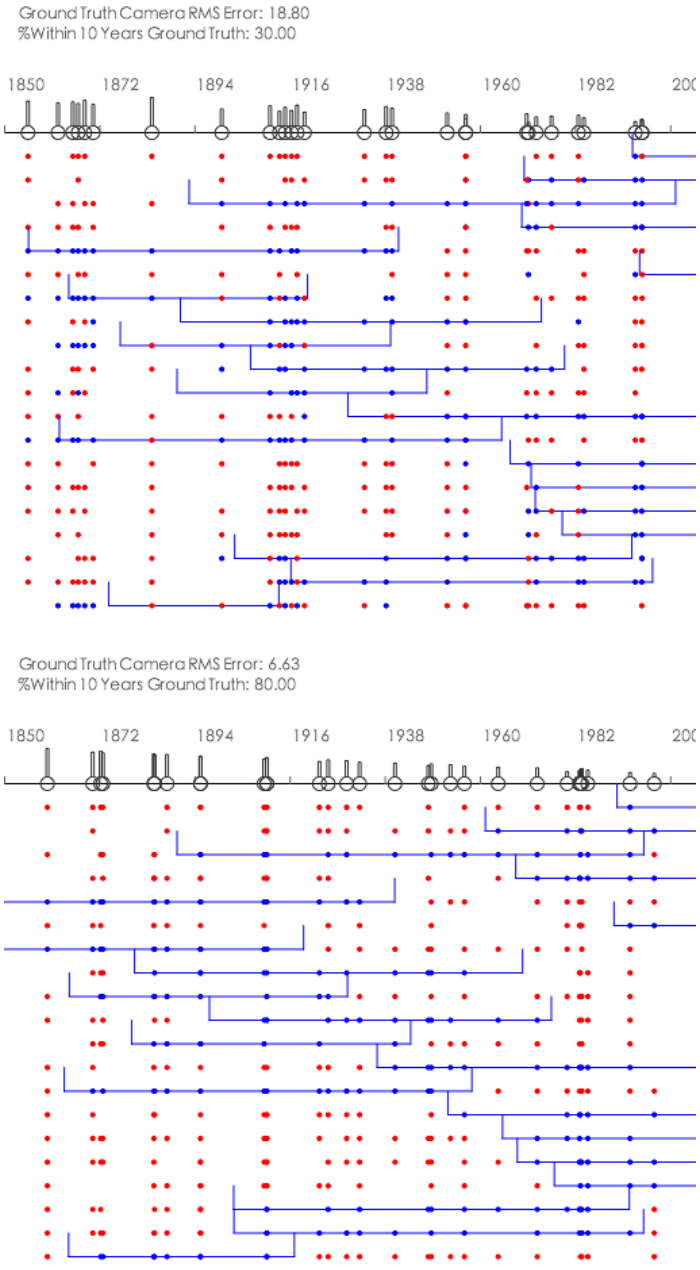


Figure 47: Continuous Optimization on a Synthetic Scene. For a synthetic scene consisting of 30 images observing 20 buildings, our continuous temporal optimization method is able to reduce the RMS error on image dates from 18.8 to 6.63 years based an initialization with just 2 of the 30 images fixed to their correct date. In the above figure, black circles at the top indicate camera dates, blue and red dots indicate positive and negative observations on buildings, and blue bars indicate time intervals for buildings.

Unfortunately, for our real scene, continuous optimization does not produce good results. Note that continuous optimization relies upon an initial model estimate and can take advantage of constraints on image dates, so we can now combine all three methods described in this chapter. Given known dates for only 19% of the city model images, we first use structural visibility dating to initialize the dates for each undated image and the date ranges for each structure. Starting from an initial RMS error of 27.84 years from structural visibility dating, continuous optimization alone reduces this RMS error to 27.30 years, only a small reduction and not the result for which we initially hoped. Even when we add appearance-based matching to get a distribution over each image's date (which takes the form of additional error terms during the continuous optimization process) performance is not improved.

To understand the relatively small improvement brought about by continuous optimization on our real data, we run a series of tests on synthetic data. We examine the effect of the sparsity of the visibility matrix on continuous optimization behavior (i.e. the fraction of entries in the visibility matrix which contain neither a positive nor a negative observation). If we had an omni-directional camera in an occlusion-less environment, then for each image, we would know for certain whether each structural element was present or absent, leading to a completely dense visibility matrix. In practice, cameras have limited fields of view, and occlusions do block observations, leading to sparse observation matrices. We find in our experiments that in the presence of a dense visibility matrix, continuous optimization routinely converges to an optimal solution, but that this property breaks down between 70% and 80% sparsity. The visibility matrix used in our real experiments (pictured in Figure 44) has a sparsity of 81.7%, suggesting that there may be fundamental problems inherent to the scene and the cameras used to capture it, which make it unsuitable for use with the continuous optimization framework we have developed here.

Based on these experiments with three different absolute dating methods, we can conclude that structural visibility dating performs extremely well when most image dates are

known, that appearance-based dating is very promising (especially when using images of the same city), but that our continuous optimization method, at least in its current form, is not well-suited to working with real data.

5.7 Discussion

This chapter has focused on adding absolute dates to the temporal inference problem. In the structural visibility dating methods described above, the human effort required to perform manual point correspondence and structure grouping for occlusion geometry is not insubstantial. We would like to find automatic methods for both correspondence and building construction. However, we also note that the current alternatives involve expert historians and archivists intimately familiar with an individual city's history and architecture, and can be equally labor intensive.

In addition, we expend computational effort estimating the date intervals of buildings while there may exist historical records that detail this information. The advantage of the presented method is that it requires no outside information, and is therefore computable purely from the widely available collections of dated and undated historical photographs – this self-containment means that non-expert humans can be used to perform the task currently and opens the possibility of full automation in the near future.

Finally, we note that the mere process of identifying point correspondences across historical images is by its nature more rigorous than the usual process applied to locate and date archival photographs. As such, during the building of the model used above, several inconsistencies in both location and date were noted between supposed ground truth historical image annotations and the results of structure from motion with structural dating. Upon inspection, the computed results appear to be more correct in several cases, suggesting the above techniques as a useful method for both generating and verifying a consistent historical record, even if manual effort is required.

5.8 Conclusion

Large collections of historical and archival photographs are quickly becoming available online, many with known dates and many which are undated. We have demonstrated the first known approach to the problem of automatically dating urban photographs via global appearance and 3D structure. Specifically, we have presented a method of assigning dates to unlabeled urban photographs with three primary contributions: (1) a structural visibility dating method which uses structure from motion to transfer dates from dated to undated photos, (2) an appearance-based dating method which matches input images to a large database of dated historical images, and (3) a continuous optimization framework which combines structural visibility and appearance-based date estimation techniques while pointing the way toward a complete solution to the temporal inference problem.

5.9 Connections to a Probabilistic Framework

The methods explored in this chapter serve as a stepping stone to the probabilistic temporal inference framework we present in the next chapter. Importantly, the results of our experiments with these methods reveal shortcomings and suggest the way to this ultimate Bayesian formulation of the problem. Though an optimization method capable of estimating absolute dates (not just orderings of images) is desirable, the chosen continuous nonlinear optimization method presented here has several drawbacks: a bad initialization will lead to failure to find the proper solution which minimizes the error, the error minimization formulation is ad-hoc and lacking some theoretical justification, and the function over which we are optimizing is only differentiable if we force it be so by taking the expectation of the observed value with respect to logistically distributed end-points of our structure intervals – we prefer a method which flows more naturally from the problem formulation.

The method underlying the structural visibility dating method is preserved in a probabilistic framework presented in the next chapter. Rather than observed buildings putting hard limits on the range of possible dates for an image, the new framework will return a

probability distribution on the date of each image in an MCMC framework. The decision in this chapter to reason about buildings rather than individual points is reasonable, but the reliance on manually measured points and manually constructed buildings may limit the usefulness of this method, so we introduce an automated solution to segmenting observed 3D points into building-like objects.

Finally, rather than using appearance matching as the primary information about an uncertain image's date, we make use of all available temporal information, including uncertain date labels like "circa 1930." However, we make room for appearance matching as a source of information in our new probabilistic framework, such that any improved results using this method can still be taken advantage of in our complete framework.

Chapter VI

PROBABILISTIC TEMPORAL INFERENCE FRAMEWORK

In this chapter, we develop a *Bayesian probabilistic formulation of the temporal inference problem*, improving the visibility reasoning introduced in Chapter 4 by using the *time-varying geometry* of the scene as the occlusion geometry, incorporating *uncertain date information* about images and relying on a probabilistic variant of structural visibility dating as in Chapter 5. By adopting this probabilistic framework, we avoid the requirement of a complete and accurate set of observations of each object in each image, enabling us to reason about *automatic 3D reconstructions* containing hundreds of thousands of points.

6.1 Introduction

Recent progress in 3D reconstruction from images has enabled the automatic reconstruction of entire cities from large photo collections (Agarwal et al., 2009), and yet these techniques largely ignore the fact that cities can change drastically over time. In this chapter, we introduce a language for representing time-varying structures, and a probabilistic framework for doing inference in these models. The goal of this framework is to enable the recovery of a date for each image and a time interval for each object in a reconstructed 3D scene.

As institutions digitize their archival photo collections, millions of photographs from the late 19th and 20th centuries are becoming available online, many of which have little or no precise date information. Recovering the date of an image is therefore an important task in the preservation of these historical images, and one currently performed by human experts. In addition, having a date on every image in a 3D reconstruction would allow for intuitive organization, navigation, and viewing of historical image collections registered to 3D city models. Discovering the time intervals of existence for every object in a scene is also an essential step toward automatically creating *time-varying* 3D models of

Figure 48: We build a 3D reconstruction automatically from images taken over multiple decades, and use this reconstruction to perform temporal inference on images and 3D objects. The left image was taken in 1956 while the right photo was captured in 1971 from nearly the same viewpoint.

cities directly from images. Toward this end, we introduce a probabilistic framework for performing temporal inference on reconstructed 3D scenes.

6.1.1 Related Work

A number of recent approaches to large-scale urban modeling from images have produced impressive results (Pollefeys et al., 2008; Agarwal et al., 2009; Zebedin et al., 2008), though none have yet dealt explicitly with time-varying structure. In (Snavely et al., 2006), a historical Ansel Adams photograph is registered to a reconstructed model of Half Dome in Yosemite National Park, but there is no notion of time in this process – only the location of the image is recovered. Additionally, since we are dealing with historical photographs,

approaches that rely on video (Pollefeys et al., 2008), densely captured data (Zebedin et al., 2008), or additional sensors are not directly applicable to our problem.

Current *non-automated* techniques for dating historic photographs include identifying clothing, hairstyles, and cultural artifacts depicted in images (Moorshead, 2000; Pols, 2002), and physical examination of photographs for specific paper fibers and chemical agents (Messier, 2005). Our approach deals with digitized photographs and contain few human subjects, so we instead opt to reason about the existence and visibility of semi-permanent objects in the scene.

Visibility and occlusion reasoning have a long history in computer vision with respect to the multi-view stereo problem (Kang et al., 2001; Kutulakos and Seitz, 2000). A space carving approach is used in (Kutulakos and Seitz, 2000) to recover the 3D shape of an object from multiple images with varying viewpoints. This involves reasoning about occlusions and visibility to evaluate the photo-consistency of scene points, and relies upon the assumption that the space between a camera center and a visible point is empty. More recently in (Furukawa et al., 2009), visibility is used to provide evidence for the emptiness of voxels in reconstructing building interiors. Our visibility reasoning approach differs from all of these in that both the potentially visible objects and potentially occluding objects vary with time, thus invalidating all the visibility assumptions that apply to static scenes. In our approach, we will be searching for a temporal story that explains why we do and do not see each object in each image.

In Chapter 4 on relative temporal inference, we proposed a constraint-satisfaction method for determining temporal *ordering* of images based on manual point correspondences. This approach suffers from a number of weaknesses: only an image ordering is recovered, there is no way to incorporate known date information, the occlusion model is static, manual correspondences are required, and there is no concept of objects beyond individual points. In contrast, our approach offers a number of advantages:

Time-Dependent Occlusion Geometry. A major problem with our relative temporal

inference method is the assumption of a fixed set of occluding geometry. Here, we treat the *uncertain scene geometry itself* as the occlusion geometry, which complicates visibility reasoning but which is necessary for dealing with real-world scenes.

Continuous, Absolute Time. Our absolute temporal inference method recovers a specific continuous date and time for each image and is able to explicitly deal with missing and uncertain date information while incorporating known dates into the optimization problem. Relative temporal inference only deals with orderings of images.

Automatic 3D Reconstruction. The manual correspondences in Chapter 4 act as perfect observations, which are not present in an automatic reconstruction. Automated feature matching cannot ensure that every feature is detected in every image, so we must deal with missing measurements.

Object-Based Reasoning. Rather than reasoning about the visibility of points as in Chapter 4, we reason about entire 3D objects which can be composed of numerous points, or any other geometric primitives. Crucially, each object explicitly has its own time interval of existence.

In addition, the method of Chapter 4 turns out to be a special case of our more general probabilistic framework. Through developing this new probabilistic temporal inference framework, we simultaneously gain insight into the previous relative temporal inference approach while creating a more powerful method for reasoning about temporal information in reconstructed 3D scenes.

6.2 Approach

The traditional Structure from Motion (SfM) problem is concerned with recovering the 3D geometry of a scene and of the cameras viewing that scene. In this work, in addition to this *spatial* information we are also interested in recovering *temporal* information about the scene structure and the cameras viewing the scene. This temporal information consists of a date for each camera and a time interval for each 3D point in the scene. Though we can

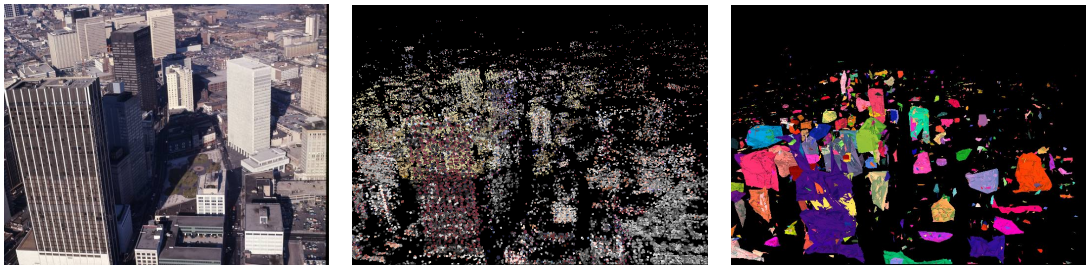


Figure 49: Point Groupings. The 3D points that result from Structure from Motion are unsuitable for use in visibility reasoning because (1) they are not reliably detected in every image, (2) they do not define solid occlusion geometry, and (3) there are too many of them. We solve all these problems by grouping 3D points into the objects about which we will reason. Points which are physically close and have been observed simultaneously in at least one image are grouped into these larger structures.

theoretically solve for both the spatial and temporal SfM parameters simultaneously, *we choose here to decompose the problem into two steps*, first solving traditional SfM (Section 6.4.1) and then solving the temporal inference problem (Section 6.3).

6.2.1 Time-Varying Structure Representation

We first define the representation we will use to perform temporal inference on reconstructed 3D scenes. To do so, we expand upon the representation of temporal parameters used in the previous chapter. Given a set of n images $I_{1..n}$ registered to a set of m 3D objects $O_{1..m}$, we wish to estimate a time t associated with each image, and a time interval (a,b) associated with each 3D object. We represent the entirety of these temporal parameters with $T = (T^O, T^C)$ where

$$T^O = \{(a_i, b_i) : i = 1..m\}$$

is a set of time intervals, one for each object, and

$$T^C = \{t_j : j = 1..n\}$$

is a set of time-stamps, one for each image.

We assume that we are given a set of geometric parameters $X = (X^O, X^C)$ for the scene, where $X^O = \{x_i : i = 1..m\}$ describes the geometry of each object and $X^C = \{c_j : j = 1..n\}$ describes the camera geometry for each image. The approach is general and these 3D

Figure 50: Uncertain Image Dates. While some historical images have known dates, a large number are labeled as "circa" a given year to indicate uncertainty in the estimated image date, and some have no date information at all. These images from the Atlanta History Center are labeled "circa 1910" (left), "circa 1955" (middle), and "undated" (right).

objects can be, for example, points, planes, or polygonal buildings. The only requirement is that each 3D object must be detectable in images and must be capable of occluding other objects.

6.2.2 Sources of Temporal Information

In this work, we assume that for *some* images we have at least uncertain temporal information. Without any time information, the best we can do is determine an ordering as in Chapter 4 on relative temporal inference. In practice, we will usually have a mix of dated images, undated images, and images with uncertain date estimates.

Modern digital cameras nearly always embed the precise date and time of the photograph in the Exif tags of the resulting image file. This includes the year, month, day, hour, minute, and second at which the image was captured. Thus, we have nearly a decade of time-stamped digital photos compared to the previous 17 decades of photography which lacks this precise temporal information. Digitized historical photographs will have associated date information only when a human archivist manually enters such a date into a database. When available, precise dates can be found in the original photographer's notes, but the more common case is that a human exercises judgment to place a date label like "circa 1955" on the photograph (see Figure 50).

We examined the date information on a set of 337 historical images from the Atlanta History Center (see Figure 51) and found that less than 11% of the images have a known

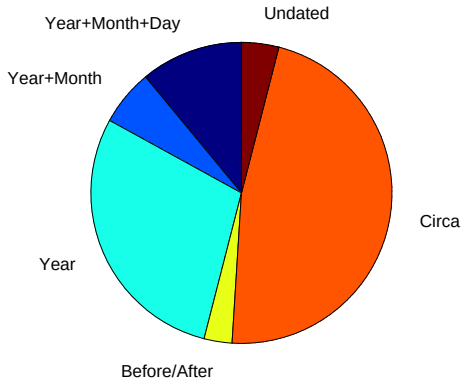


Figure 51: Image Date Information. For a set of 337 historical images of Atlanta, less than 11% of the images have a known year, month, and day, 47% are "circa" some year, 29% have a known year, 6% have a known year and month, 3% are "before" or "after" some year, and 4% are completely undated. This lack of precise temporal information for a majority of historical photographs motivates our work.

year, month, and day. Of all images, 47% are "circa" some year, 29% have a known year, 6% have a known year and month, 3% are "before" or "after" some year, and 4% are completely undated. This lack of precise temporal information for a majority of historical photographs motivates our work.

6.3 Probabilistic Temporal Inference Model

Our goal is to estimate the time parameters T of a set of images and objects given the geometric parameters X of a reconstructed 3D scene. In addition, we assume that we are given a set of observations $Z = \{z_{ij} : i = 1..m, j = 1..n\}$ where each z_{ij} is a binary variable indicating whether object i was observed in image j. In what follows, we will be searching for the set of temporal parameters T that best explain the observations Z, telling us why we see certain objects in some images but not in others. In Bayesian terms, we wish to perform inference on all temporal parameters T given observations Z and scene geometry X,

$$P(T|Z,X) \propto P(Z|T,X)P(T) \tag{1}$$

In the following two sections, we discuss the likelihood term $P(Z|T,X)$ first and then the prior term $P(T)$.

6.3.1 Observation Model

The key term which we need to evaluate is the likelihood $P(Z|T,X)$. Because the observations are conditionally independent given T, we can factor the likelihood as:

$$P(Z|T,X) = \prod_{z_{ij} \in Z} P(z_{ij}|T,X) \tag{2}$$

This is the product, over all objects in all images, of the probability of each individual observation z_{ij} given T and X. Evaluation of the terms $P(z_{ij}|T,X)$ relies on three factors:

Viewability: Is object i within the field of view of camera j? This only depends on the geometry X, more specifically for each measurement z_{ij} we can deterministically evaluate the function $InFOV_{ij}(X)$ that depends only on the object and camera geometry x_i and c_j.

Existence: Did object i exist at the time image j was captured? This only depends on the temporal information T, as given T we can deterministically evaluate the functions $Existence_{ij}(T) = a_i \leq t_j \leq b_i$.

Occlusion: Is object i occluded by some other object(s) in image j? This attribute, $Occluded_{ij}(T,X)$, depends on both temporal information T and geometry X. Specifically, $Occluded_{ij}(T,X)$ depends upon *all* time intervals T^O, *all* object geometry X^O, and camera parameters (t_j, c_j).

Below we discuss each of these factors in turn.

6.3.1.1 Viewability

Based on viewability alone, we can factor the likelihood (2) in two parts: one that depends on the temporal information T and one that does not. Indeed, if we define the *viewable set* $Z_V = \{z_{ij}|InFOV_{ij}(X)\}$, we have

$$P(Z|T,X) = k \prod_{z_{ij} \in Z_V} P(z_{ij}|T,X) \tag{3}$$

where k is a constant that does not depend on T, and hence is irrelevant to our inference problem. In practice all the measurements z_{ij} not in the viewable set Z_V are 0, so the above simply states that we do not even need to consider them. However, the viewability calculation has to be done to be able to know *which* measurements z_{ij} to disregard.

6.3.1.2 Existence

The viewable set Z_V can, given the temporal information T, be further sub-divided into two sets Z_N and Z_P, where $Z_P = \left\{ z_{ij} | z_{ij} \in Z_V \wedge Existence_{ij}(T) \right\}$ corresponds to the set of image-object pairs (i, j) that co-exist given T, and its complement $Z_N = Z_V \setminus Z_P$ is the set of all measurements predicted to be negative because the object and image did *not* co-exist. *Crucially, note that this division depends on the temporal parameters T.* Hence, the likelihood (3) can be further factored as

$$P(Z|T,X) = k \prod_{z_{ij} \in Z_N} P_N(z_{ij}) \prod_{z_{ij} \in Z_P} P_P(z_{ij}|T,X)$$

The first product above dominates the likelihood, as it is very improbable that an object i will be reported as visible in camera j if in fact it did not exist at the time image j was taken. In other words, $P_N(z_{ij} = 1) = \rho$, with the *false positive probability* ρ a very small number. Hence the likelihood stemming from the observations in Z_N is simply

$$P(Z_N|T,X) = \prod_{z_{ij} \in Z_N} P_N(z_{ij}) = \rho^{FP}(1-\rho)^{CN} \tag{4}$$

where FP and CN are the number of *false positives* and *correct negatives* in the set Z_N, with $FP + CN = |Z_N|$. Note that in the case $\rho = 0$ the likelihood $P(Z_N|T,X)$ evaluates to zero for any assignment T violating an existence constraint.

6.3.1.3 Occlusion

Finally, if object i *does* exist when image j is taken, then the probability $P_P(z_{ij}|T,X)$ that it is observed depends upon whether it is occluded by other objects in the scene, i.e.,

$$P_P(z_{ij}|T,X) = \eta \times P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O) \tag{5}$$

with η the *detection probability* for unoccluded objects. Since we rely on SfM algorithms, even unoccluded objects might not be detected properly: the reasons include failure during feature detection or matching, or occlusion by an un-modeled object such as a tree or car. Although we use a constant term η here, this probability could be evaluated on a per object/per image basis using the known scene and camera geometry. For example, we could capture the notion that a small object is unlikely to be observed from a great distance despite being in the field of view.

The occlusion factor $P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O)$ can in turn be written as the probability of object i not being occluded by any other object k,

$$P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O) = \prod_{k \neq i}(1 - P(Occlusion_{ijk}|t_j, c_j, a_k, b_k, x_k, x_i))$$

where $Occlusion_{ijk}$ is a binary variable indicating whether or not object i is occluded by object k in image j. The probability $P(Occlusion_{ijk}|.)$ can vary from 0 to 1 to account for partial occlusions of objects. With this model, the overall probability $P(Occluded_{ij}|t_j, c_j, T^O, X^O)$ that object i has been occluded by *something* in image j *increases* as more individual objects k partially occlude object i. A specific occlusion model will be discussed further in Section 6.4.3.

6.3.2 Temporal Prior

The term $P(T)$ in Equation (1) is a prior term on temporal parameters. This can be further broken down into image date priors $P(T^C) = \prod_{j=1..n} P(t_j)$ and object time interval priors $P(T^O) = \prod_{i=1..m} P(a_i, b_i)$.

If we have any prior knowledge about when an image was taken, we account for it in the individual $P(t_j)$ prior terms. We may know an image's time down to the second, we may just know the year, or we may have a multi-year estimate like "circa 1960". In all such cases, we choose a normal distribution $P(t_j) = N(\mu, \sigma^2)$ with a σ appropriate to the level of uncertainty in the given date. When we have no date information at all for a given

image, we use a uniform distribution appropriate to the data set – for example, a uniform distribution over the time between the invention of photography and the present. Though not used here, object interval priors $P(a_i, b_i)$ can also be chosen to impose an expected duration for each object.

6.3.3 Framework Extensions

An added benefit of this probabilistic temporal inference framework is that it becomes easy to extend the model to account for additional domain knowledge (though we do not use these extensions here). We can introduce a term $P(X^O|T^O)$ which encodes information about the expected heights of buildings given their construction dates, exploiting the fact that buildings have gotten progressively taller at a known rate over the last century, or a term $P(X^C|T^C)$ which incorporates prior information on the expected altitude of cameras given image dates, again exploiting the fact that we have records describing when airplanes, helicopters, and tall rooftops came into being and enabled higher-altitude photographs to be captured. Both of these extensions would require the measurement of a known object to be specified in the scene in order to reason in non-arbitrary units.

Finally, we can introduce a term $P(I|T^C)$ specifying a distribution on image features for photos captured at a given time. Such features might include color or texture statistics, or even detections of cultural artifacts like cars or signs which are typical of specific historical eras, properties which already allow humans to roughly estimate the date of a photograph of an unfamiliar city scene. This would be especially significant in the case of historic cities which have not structurally changed much during the era of photography, where visibility reasoning alone may not be sufficient to pinpoint the date of an image. Though we explored this kind of approach in Chapter 5, we do not make use of this method in the experiments described here.

6.3.4 Temporal Inference Algorithms

We are interested in finding the the optimal value T^* for the temporal parameters according to the maximum a posteriori (MAP) criterion:

$$T^* = \underset{T}{\text{argmax}} \;\; P(T|Z,X)$$

Observe that, based on the above formulation, given a hypothesized set of temporal parameters T we can directly evaluate Equation (1) to get the probability of the hypothesized time parameters. Therefore, we perform temporal inference by *sampling* time parameters to find those that maximize the probability of the data.

6.3.4.1 Markov Chain Monte Carlo

We adopt a Markov Chain Monte Carlo (MCMC) approach to draw samples from the posterior distribution $P(T|Z,X)$ in order to find the optimal set of parameters T^*. Following the Metropolis-Hastings (Hastings, 1970) algorithm, we start from an initial set of temporal parameters T and propose a move to T' in state space by changing one of the t_j, a_i, or b_i values according to a proposal density $Q(T';T)$ of moving from T to T'. We accept such a move according to the acceptance ratio:

$$\alpha = \min\left\{\frac{P(T'|Z,X)\,Q(T;T')}{P(T|Z,X)\,Q(T';T)}, 1\right\} \tag{6}$$

Our proposals involve randomly choosing a time parameter and adding Gaussian noise to its current value, such that our proposal distribution is symmetric, and the acceptance ratio is simply the ratio of the posterior probability $P(T|Z,X)$ of each set of temporal variables. Following this approach, we draw samples from the posterior probability $P(T|Z,X)$, keeping track of our best estimate for T^* as we do so.

We make this sampling approach more efficient by sampling only on image dates T^C, and analytically solving for the optimal object time intervals T^O for a given configuration of T^C. To do so, we note that the dominant likelihood part given by Equation (4) factors

over objects i:

$$\prod_{z_{ij} \in Z_N} P_N(z_{ij}) = \prod_i \left\{ \prod_{j|z_{ij} \in Z_N} \rho^{FP_i} (1-\rho)^{CN_i} \right\}$$

Given the image dates T^C, we can eliminate false positives FP_i for each object i by setting

$$a_i \leq \min \left\{ t_j | z_{ij} = 1 \right\} \text{ and } b_i \geq \max \left\{ t_j | z_{ij} = 1 \right\}$$

In other words, and obvious in hindsight, *we make each object's interval such that it starts before its first "sighting" and ends after its last "sighting"*. In practice we found that extending the intervals beyond the minimum range indicated above has a negative effect on the solution: while extending an interval can help "explain away" negative observations of other objects, this also automatically incurs a $(1-\eta)$ likelihood penalty for every image in which the object is now not observed. This dominates the potentially beneficial effects.

Hence, for every proposed change to the image dates T^C, we adapt the object intervals (a_i, b_i) to minimize the existence constraints (4). This changes the set Z_P for which the occlusion/detection likelihood (5) needs to be evaluated. It is computationally efficient to propose to only change one image date t_j at a time, in which case only objects in view of camera j have their intervals adjusted, and calculating the acceptance ratio (6) is easier. However, occlusion effects will still have non-local consequences: in Section (6.4.3) we discuss how to deal with those efficiently as well.

6.4 Implementation

The above formulation is a general temporal inference framework applicable to a variety of situations. For the specific case of reasoning about cities over decades of time, we must specify how we recover geometry X using SfM and what kind of objects O we are dealing with, as well as how these objects are detected and how they occlude each other.

6.4.1 Structure from Motion

Before performing any temporal inference, we run traditional SfM to recover the camera geometry X^C and a set of 3D points which will form the basis for the geometry of our 3D

objects X^O. For this purpose, we use the Bundler SfM software from Snavely (Snavely, 2008) with SIFT implementation from VLFeat (Vedaldi and Fulkerson, 2008). Depending on the connectivity of the match table, there may be multiple disconnected reconstructions that result from this SfM procedure. In our case, we are not interested in the reconstruction with the largest number of images, but rather the one containing images which *span the largest estimated time period.*

6.4.2 Object Model

We must define the set of 3D objects $O_{1..m}$ on which to perform temporal inference. The output of SfM is a large number of 3D points, but in a large-scale urban reconstruction, it makes more sense to reason directly about 3D buildings than 3D points. Segmenting point clouds into buildings is a difficult task, complicated here by the fact that *multiple buildings can exist in the same location* separated only by time. To solve this problem, we perform an oversegmentation of the points into point-groups, analogous to superpixels used in 2D segmentation (Ren and Malik, 2003). Specifically, if two 3D points are closer than a threshold d_{group} and are also observed simultaneously in at least N_{group} images, we link them together and then find connected components among all linked points (see Figure 49).

Grouping points in this way leads to several benefits. First, we can count an observation of any one point in a group as an observation of the whole group (see Figure 52). This increases the chance of successfully detecting each object in as many images as possible, reducing false negatives. By reducing the number of 3D objects, we also vastly reduce the computational burden during occlusion testing. For the purposes of visibility reasoning, we triangulate each group of points (based on either a 3D convex hull or a union of view-point specific Delaunay triangulations) and use this triangulated geometry to determine which groups potentially occlude each other.

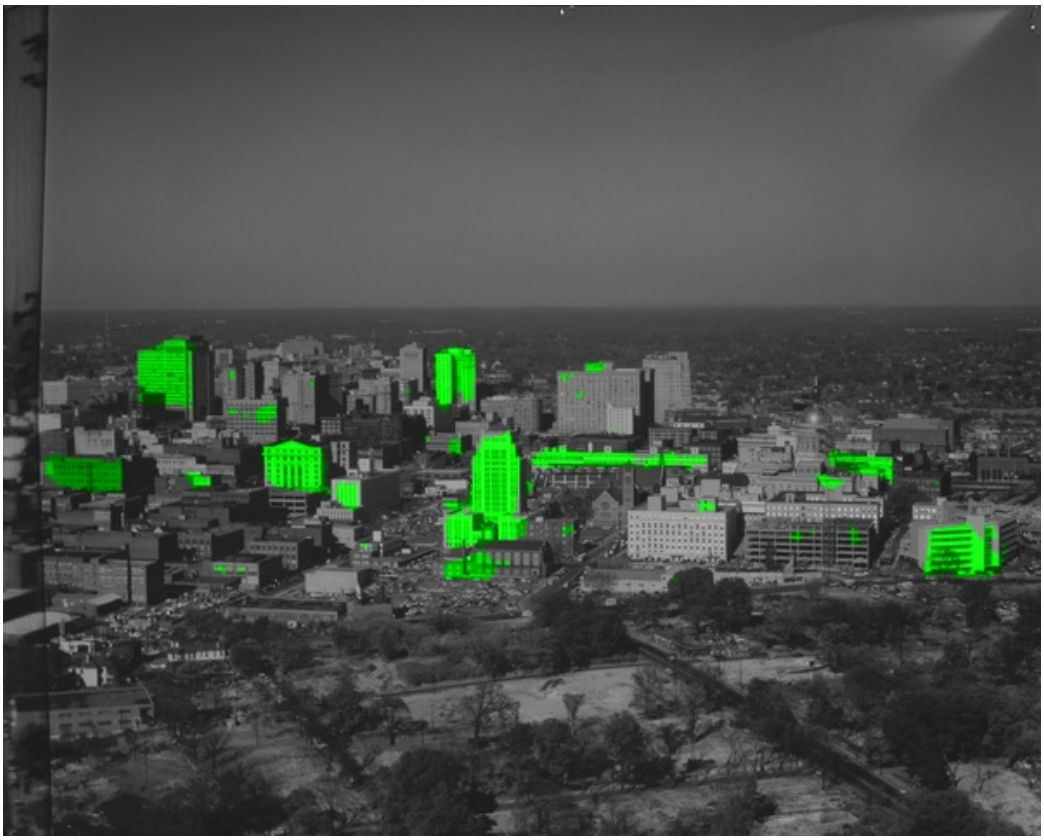


Figure 52: Object Observations. Our framework reasons about observations of 3D objects in images. We group the 3D points from SfM into larger structures and count the detection of at least one point in the group as an observation of the entire structure. Regions highlighted in green (above) represent observed objects in this image. False negative observations are undesirable but unavoidable, and we account for them in our probabilistic framework.

6.4.2.1 *Building Models via Convex Hulls and Ground Plane Estimation*

In the present work, we are also interested in creating *building* models, not just generic objects, to stand in as occluders, to link together observations on multiple points, and for user interaction with 4D city models. There is a large body of work on building architectural models from images (Bauer et al., 2003; Dick et al., 2002; Mueller et al., 2007; Cipolla et al., 1999). We introduce a method to create building-like models from point groups, inspired by our method of interactively creating solid building geometry by extruding a polygon of connected roof points down to a ground plane. To achieve the same result automatically, we combine two techniques: 3D convex hulls, and ground plane estimation.

Given a set of points, we compute a 3D convex hull as the smallest convex polytope containing all the points. When we compute the convex hulls of all point groups in a scene, we generally wind up with a set of polygonal object models that appear to "float" above the ground – this is because SfM does not reconstruct 3D points on the lower levels of many buildings for which we only see the upper floors. To correct this effect, we project all points in a given set onto the ground plane, and take the convex hull of a new set of points including both the original points and the ground-projected points. The result is a building model that extends to the ground, and is an improved representation both for occlusion modeling and visualization.

We automatically estimate the ground plane by computing the eigenvectors of the covariance matrix of all camera centers in the 3D reconstruction. We make the assumption that most images are taken from the ground, and therefore fitting a plane to the recovered camera locations gives a reasonable ground plane estimate, even if this assumption is not strictly true.

6.4.3 Occlusion Model

We must determine which objects in our scene potentially occlude which other objects, as this information plays a pivotal role in evaluating the probability of a given configuration of

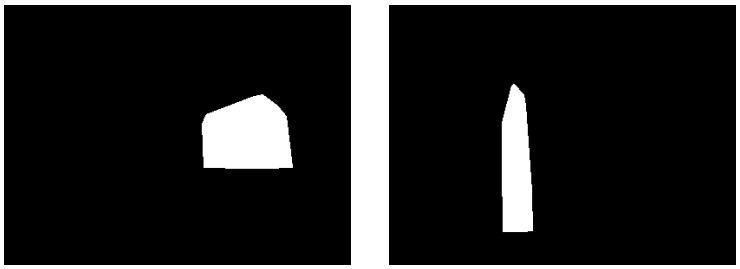


Figure 53: Occlusion Computation. Binary images are used to quickly decide which 3D structure points are potentially occluded by each object. For each of m objects in n images, we render just the single object's triangles as white on a black background. Only if a point's 2D projection lands on a white pixel in a given image should further depth tests be conducted to determine whether the object truly occludes the point in that image. Because 99.9% of points land in black regions, this offers enormous computational savings.

temporal parameters as described in Section 6.3.1.3. This involves the creation of an occlusion table, a three-dimensional table of size $m \times m \times n$ which specifies, for each image, the probability $P(Occlusion_{ijk}|X,T)$ that object k occludes object i in image j if *both objects exist at the same time*. The occlusion table is extremely sparse, but it is the most expensive computation in the entire algorithm due to the fact that m^2n geometric calculations must be made to compute it.

This expensive occlusion table computation is where we pay the price for not committing to a static set of occlusion geometry as in Chapter 4. As our model's time parameters vary during optimization, the number of unique occlusion scenarios is 2^m where the number of objects m reaches into the thousands. We cannot precompute occlusion information for all these scenarios, nor do we want to compute occlusion events on the fly while evaluating the probability of a specific set of temporal parameters – this slows down evaluation by an order of magnitude.

Occlusion Computation As described above, we have a list of 3D triangles associated with each object for occlusion purposes. Rather than explicitly computing ray-triangle intersections between each camera center and each structure point for every triangle in the occlusion geometry as in Chapter 4, we use an image-space approach. We first render a binary image (see Figure 53) for each object in each camera – despite the large number

of rendered images ($m \times n$) this is a very fast operation either on the GPU or in software. Each image is white where the potentially occluding object's triangles project into the image and black everywhere else. By projecting each 3D structure point into each image, we can quickly detect potential occlusion events by examining the pixel color at the projected location of each point. If a point projects onto a white pixel, further depth tests are performed to determine occlusion, but in our experiments greater than 99.9% of points project onto black background pixels, which means no further tests are necessary, saving enormous computation.

So far, we have computed *point-object* occlusion events. To compute *object-object* occlusion probabilities, we do the following: when an object k occludes any points belonging to another object i, the probability of occlusion $P(Occlusion_{ijk}|X,T)$ is equal to the fraction of object i points which were occluded by object k.

Having pre-computed all *potential* occlusion events in this way, at run time we use the current time parameter estimate T to determine which of these occlusions actually occur at the time of each image in the model. Importantly, using this *time-dependent occlusion* approach, we can not only explain away missing observations as in the relative temporal inference method of Chapter 4, but if an object is observed when the model indicates that it should be occluded, this provides strong evidence that the occluder itself should not exist at the present time.

6.5 Relative Temporal Inference as a Special Case

Now that we have established this general temporal inference framework, we can reformulate the relative temporal inference method of Chapter 4 as a special case of the more general framework in order to understand exactly how relative temporal inference differs from our current approach. First, for the relative case, regarding temporal parameters T, all variables take on discrete values in the range $1..n$ for n images. In addition, no prior temporal information is used, so that $P(T)$ is uniform and we only need to consider the

observation model $P(Z|T,X)$.

We next examine how the constraints of the relative temporal inference problem relate to the observation model of the general framework. The constraint used in relative temporal inference is that when an object is in the field of view, exists, and is not occluded, it *must* be positively observed, or else the constraint is violated. This constraint can be encoded by altering Equation 5 above by first setting the detection probability to $\eta = 1.0$, so that there is no chance of a feature simply going undetected. Second we alter the term $P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O)$, changing it to $P(\overline{Occluded_{ij}}|c_j, X^O)$, so that it no longer depends on temporal parameters, since in the relative temporal inference framework a static geometry assumption is necessary to define which observations are *missing*. Finally, this term must be evaluated in an all or nothing manner – objects are either occluded or they are not, so that the probability of a missing observation is zero. Thus, we can summarize the observation model with the only term that has any effect on the solution: $P(z_{ij} = 0|\overline{Occluded_{ij}(X)}, InFOV_{ij}(X), Existence_{ij}(T)) = 0$. For *most* combinations of values of the attributes $Occluded_{ij}(X)$, $InFOV_{ij}(X)$, and $Existence_{ij}(T)$ in the constraint satisfaction setting, the probability of observing an object is equal to the probability of not observing it. For this reason, the relative temporal inference method has nothing to the say about the probability of observing an object given that it *is* occluded in the model, while the general probabilistic temporal inference framework naturally and correctly accounts for this case. In the CSP, there is also never a case where an object is observed outside of its time interval of existence because $Existence_{ij}(T)$ in the relative temporal inference problem is defined implicitly by always setting the structure time intervals to extend from the first positive observation of an object to the last positive observation.

Thus, we have taken the visibility reasoning of our relative temporal inference method and incorporated it into a probabilistic framework that is more general, more powerful, and more correctly models the relationships between observations and temporal parameters by

depending on a time-varying occlusion model. The addition of constraints on temporal parameters according to prior knowledge makes the framework even more powerful, finally giving us the chance to perform absolute temporal inference automatically. In the same way, the probabilistic nature of the new framework permits us to deal with noisy observations and therefore to rely on automatic correspondence and 3D reconstruction methods.

6.6 Results

We perform temporal inference experiments on both synthetic and real data. The synthetic data allows us to evaluate our method's performance with respect to ground truth. For our Atlanta data set, we will demonstrate the *successful optimization of all temporal parameters* in a scene, and for our Manhattan data, we perform leave-one-out dating experiments to show that our method can effectively *recover the dates of individual images* within the same temporal inference framework.

6.6.1 Synthetic Scene

For the synthetic scene, we have 100 images, taken over an 80 year period, observing 2112 3D points lying on the surface of 30 synthetic buildings (see Figure 54). Of these 100 images, 33% have known date, 33% are "circa" some year, and 34% have completely unknown dates. The initial date for each image is, respectively, set to its known value, rounded to the nearest decade, or uniformly sampled between 1930 and 2010. For temporal priors, we use a normal distribution with mean set to these initially estimated dates for each image, while $\sigma^2 = 10.0$ if an image is "circa" some year and $\sigma^2 = 0.001$ for known dates. Undated images have a uniform prior distribution. In all experiments, for both synthetic and real data, the proposal density for MCMC is a normal distribution with $\sigma = 50$, the detection probability is $\eta = 0.7$, and we use point-grouping parameters $N_{group} = 1$ with threshold d_{group} depending upon each scene's arbitrarily scaled geometry.

For this synthetic scene, we perform a full temporal optimization by drawing 80,000 samples of temporal parameters T using MCMC and keeping the most probable sample.

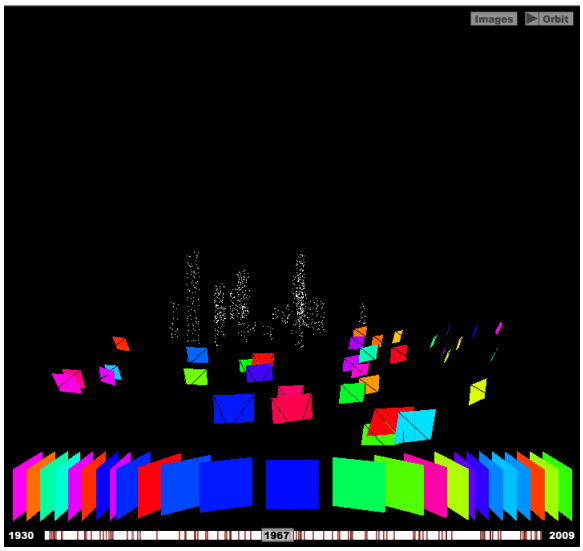


Figure 54: Synthetic Scene. We use synthetic data in order to evaluate the performance of our temporal inference method with respect to ground truth. For this synthetic scene of 100 images observing 30 buildings over 80 years, our method successfully recovers temporal information with an average error of only 2.87 years despite completely missing date information for one third of the images.

(By recording all MCMC samples, we also recover distributions over all temporal parameters rather than just a single solution, and we discuss these distributions below). To evaluate performance, we compute the root mean square (RMS) error between all estimated image dates and all ground truth image dates. For our synthetic scene, this temporal optimization procedure reduces the RMS error from 19.31 years for the initial configuration to just 2.87 years for our solution. This is an excellent result, and because the same groups of buildings persist for multiple years (making certain dates indistinguishable from others), we should not necessarily expect to be able to improve on this performance using the present method.

This synthetic scene has several properties not present in real data, including: completely accurate observations, completely accurate scene and camera geometry, and observed points evenly distributed over building surfaces (which comes into play when grouping points into objects). We can gain insight into the sensitivity of our method to changes in some of these aspects of the data by adjusting them for our synthetic scene and observing how the overall temporal inference results are affected. We choose to test what we see as

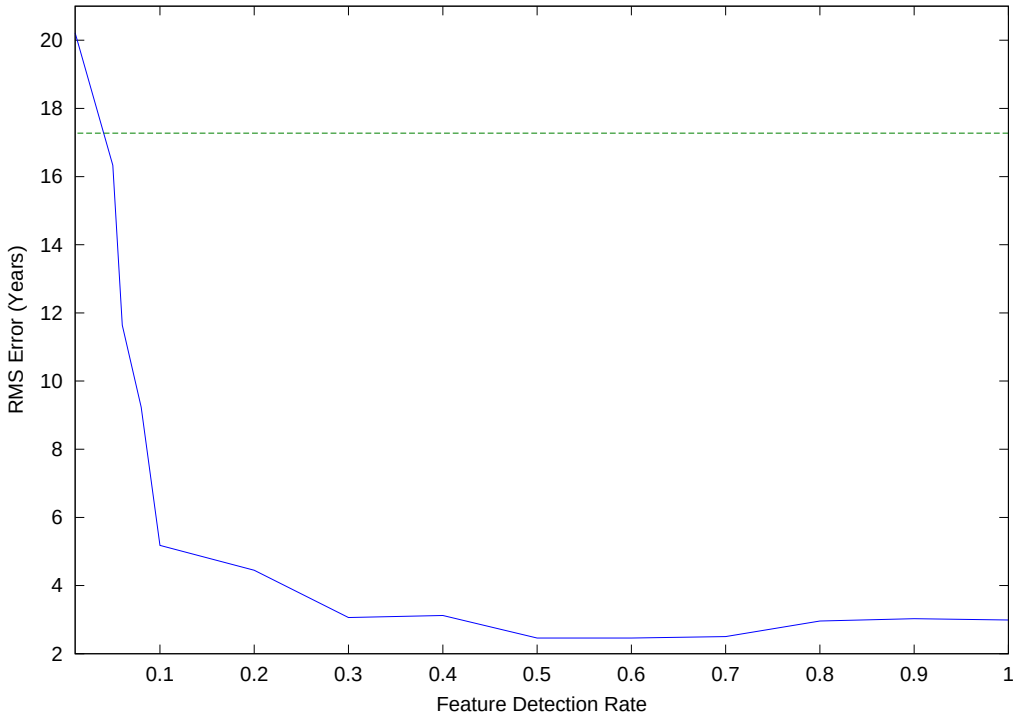


Figure 55: Feature Detection Rate. We vary the percentage of features detected for a synthetic scene and find that performance is still good with only 30% of features detected. Even beyond this point, our method degrades gracefully. The horizontal line represents the RMS error for the initial time parameters before any optimization.

the most significant parameter for real world scenes: the feature detection rate.

6.6.1.1 Feature Detection Rate

We perform an experiment where we vary the percentage of building features detected in each image of the synthetic scene. We observe that in real images, a SIFT feature corresponding to each reconstructed 3D point is not detected in every image for which the point should be visible. When we vary the detection rate for our synthetic scene from 100% to 1%, we get some surprising results (see Figure 55). At a detection rate of just 30%, the performance of our method is nearly identical to the performance for 100%. What this demonstrates is the effectiveness of grouping points into objects for the purpose

of observation. Despite the fact that 70% of the individual point observations are missing, the *buildings* are still reliably detected to perform temporal inference. We note the method gracefully degrades as observations are removed (and performance only drops significantly at lower than 10% detection). Remember that these experiments were performed with 34% of image dates completely missing. For our real data sets, we have at least approximate dates for most of the images, suggesting that even lower feature detection rates can be dealt with. This is fortunate, as the approximate feature detection rates for our real data sets are 2.7% for Atlanta and 1.5% for Manhattan. Our successful experiments on real data bear out the fact that we are able to cope with a large number of missing feature detections.

6.6.1.2 Date Distributions

Though we have presented MCMC as a means to find the most likely configuration of temporal parameters, MCMC provides us with a set of samples drawn from our target distribution. We can use these samples to recover, for each image, a *distribution* over image dates rather than just a point estimate. We visualize the temporal distributions for the 100 images of our synthetic scene in Figure 56. For the images with known dates, the density is concentrated around the true date, while for images with uncertain or unknown dates, the density is spread out. Note that in Figure 56 there are clearly several time intervals during which little building construction took place, resulting in extended time periods of equal probability for a number of images taken during these periods. In the figure, this is reflected in the wide swaths of blue color during the simulated 1940s and 1990s in our experiment.

It is clear that if no buildings are constructed or demolished during a period, then our method has no way of differentiating any specific year within the period. In real images, temporary structures like signs and billboards may act to bridge these gaps in building changes. This is a case in which additional cues from cultural artifacts like cars or clothing styles might be helpful to differentiate between different time periods, and as we discuss above, our probabilistic framework can easily accommodate terms for such information.

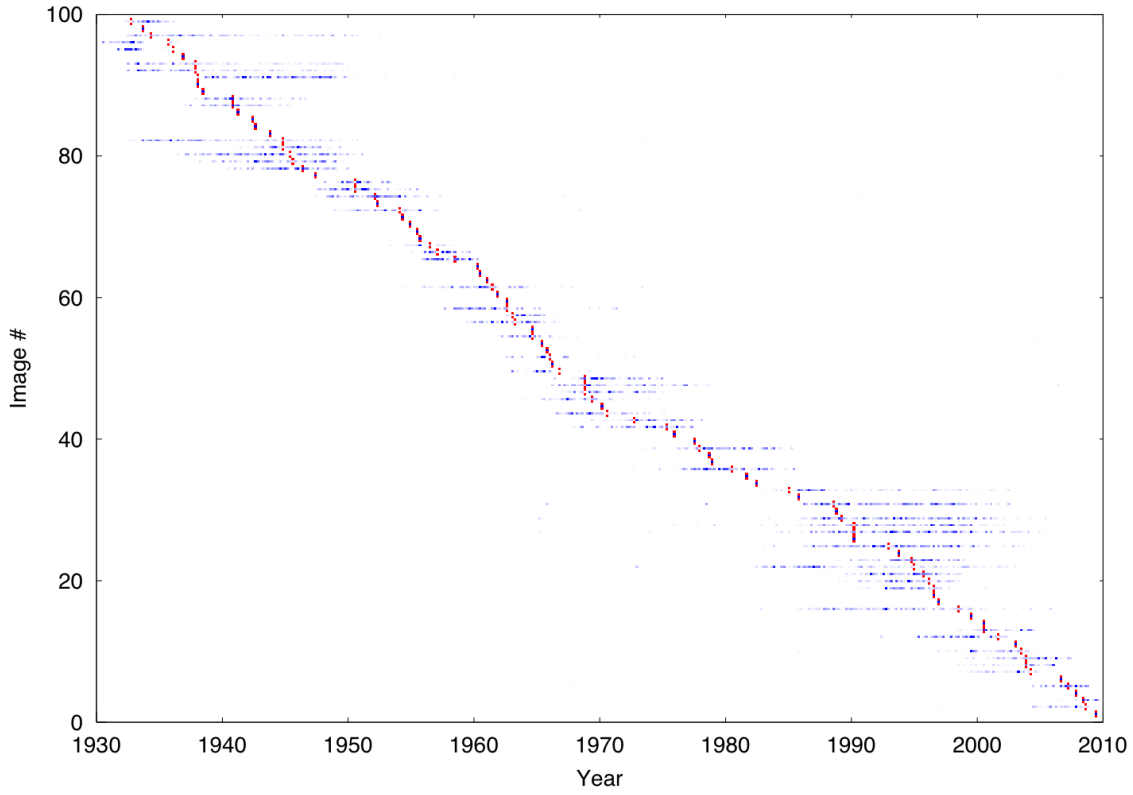


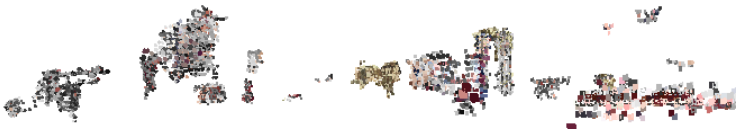
Figure 56: Synthetic Scene. Marginal date distributions for each image in the scene are represented as histograms of MCMC samples. Red pixels indicate ground truth dates, while blue pixels indicate the temporal density for each image computed over 80,000 samples.

Finally, we note that a likely reason so many real historical images are labeled "circa" some year is precisely because a human observer has been unable to narrow down a date any further than the visible structures in the scene allow. Thus, it is an *advantage* of our MCMC approach that we get not just a point estimate, but a distribution which could be construed as an interval of time during which a specific image might have been captured. This is more powerful than just an estimated date, and would serve as a useful starting point for a human expert to examine the image further.

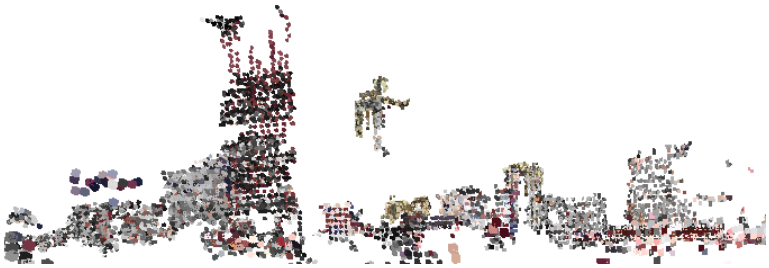
6.6.2 Downtown Atlanta

For our Atlanta data, starting from a collection of 490 images dating from the 1930s to the 2000s, the result of SfM is a set of 102 images registered to 89,619 3D points and spanning the 1950s, 1960s, and 1970s (see Figures 48 and 57). We use the above point-grouping procedure to create 3,749 objects from the original 89,619 points. We note that the largest reconstructed set of images from the input was actually a set of 127 images all taken in the 2000s, but which failed to be connected to any historic images by an automated SfM procedure. One possible cause of this problem may be that our images were not uniformly distributed across time, with a notable lack of images from the 1980s and 1990s which are not yet well-represented in either historical databases or online photo-sharing collections. We hypothesize that a denser sampling of images in both time and space would be required to link these reconstructions together. This problem, and potential solutions, are discussed further in the next chapter.

We perform a full temporal optimization to estimate all temporal parameters T for our scene, including image dates t_j and object intervals (a_i, b_i). For each image in our reconstruction, we initialized temporal parameters according to the historical date information accompanying the photographs. For temporal priors, we use a normal distribution with mean at the given date for an image, while $\sigma^2 = 0.05$ if an image is "circa" some time, $\sigma^2 = 0.01$ if given a year, $\sigma^2 = 0.001$ if given a year and month, and $\sigma^2 = 0.0001$ if a



(a) 1960

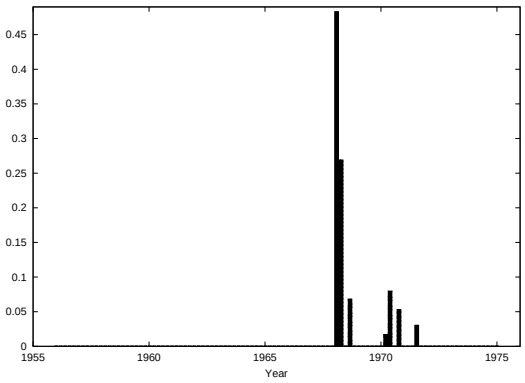


(b) 1965

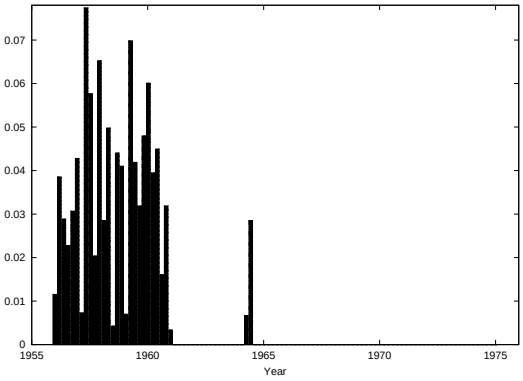
(c) 1970

(d) All Points

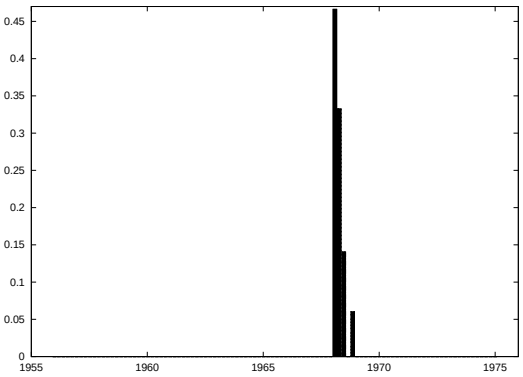
Figure 57: Object Time Intervals. By performing temporal inference, we recover a time interval for every object in the scene. Here, we use these recovered time intervals to visualize the scene at different points in time (a)(b)(c) from the viewpoint of a given photograph. In contrast, the raw point cloud (d) resulting from SfM has no temporal information.

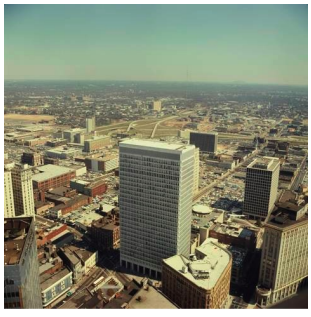


(a) Estimated Date: 1968.19 Original Date: Undated



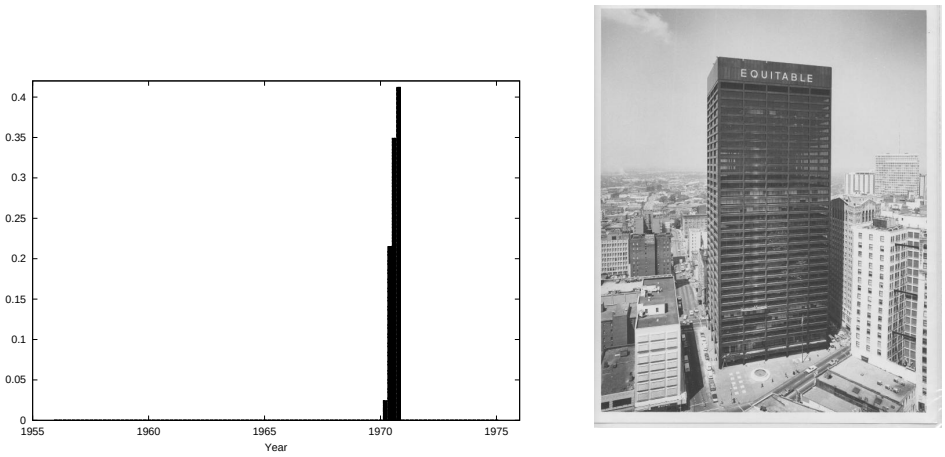
(b) Estimated Date: 1957.08 Original Date: Undated





(c) Estimated Date: 1968.24 Original Date: Circa 1967

Figure 58: Full Temporal Optimization. We simultaneously estimate all temporal parameters for the Atlanta data set and examine the resulting marginal date distributions for several images. The graphs on the left display the probability that the photo on the right was taken on a given date, computed as a histogram of samples resulting from MCMC.



(a) Estimated Date: 1970.79 Original Date: Undated

Figure 59: Result of Temporal Inference for Atlanta. For this undated image, the date distribution peaks strongly in 1970. The central building in this image was built in 1968, making this a reasonable estimate. The significance of this result is that this undated image has been integrated into a 4D model without any human intervention.

full date is specified. No temporal prior is used in the case of undated images. Note that these distributions are much tighter than the ones used for the synthetic data above, since for our real historical data we want to trust the given dates as much as possible while still allowing a better temporal solution to be found. We use the MCMC sampling procedure described above, drawing 80,000 samples to arrive at the most probable temporal solution for the entire set of 102 images in the reconstruction. On a 2.33 GHz Intel Core 2 Duo, evaluating one sample takes 0.06 seconds, so we can evaluate 1000 samples per minute. The occlusion table itself takes on average 5.5 seconds per image, and is a one-time operation totaling less than 10 minutes for this dataset. Note that actual ground truth is difficult to achieve for this historical data – any images with missing or uncertain dates have already been labeled by human experts to the best of their ability, and it is these very labels which are uncertain. Instead, we first highlight a few illustrative examples (Figures 58 and 59) to demonstrate our method's effectiveness on real-world data:

- For an image originally dated 1868 (apparently a data entry error in the historical database with the intended date of 1968), we removed this incorrect date and treated

the image as an "undated" image. The estimated date, after a full optimization of all temporal parameters in the model, was February 1968 which is precisely the year originally intended for the image. (See Figure 58 (a)).

- For another undated image (see Figure 58 (b)) the resulting marginal date distribution places samples throughout the years between 1956 and 1961. This correctly places the image before most of the city's skyscrapers were built, and the non-peaked distribution lets us capture the uncertainty in this estimate.

- An image labeled "Circa 1967" was moved up to the middle of 1968 during the temporal optimization of our model. Upon examination, this image primarily depicts a building which began construction in 1968 and another building which was demolished in 1970. While we can confirm this using building construction records, our method is able to perform this reasoning from images alone. Moreover, this demonstrates a primary strength of our probabilistic temporal inference method – that uncertain image date labels can be used to shape the temporal solution, but can also be changed when they don't agree with the observations. (See Figure 58 (c)).

- For an undated image, we get a date distribution that peaks strongly in 1970 (see Figure 59). The central building in the image was built in 1968, making this a reasonable estimate. The significance of this result is that this and other undated images have been integrated into a 4D model without any human intervention.

After performing temporal inference on all image dates and object time intervals, we visualize the results (Figure 57) by choosing a point in time and rendering only those objects which exist at this time according to the recovered time intervals. When we view the 3D reconstruction from the same viewpoint but at different points in time, the successfully recovered time-varying structure becomes clear.

In the same way that we visualized the distribution of MCMC samples for the synthetic scene, we visualize the Atlanta scene's temporal distribution in Figure 60. In this case, red

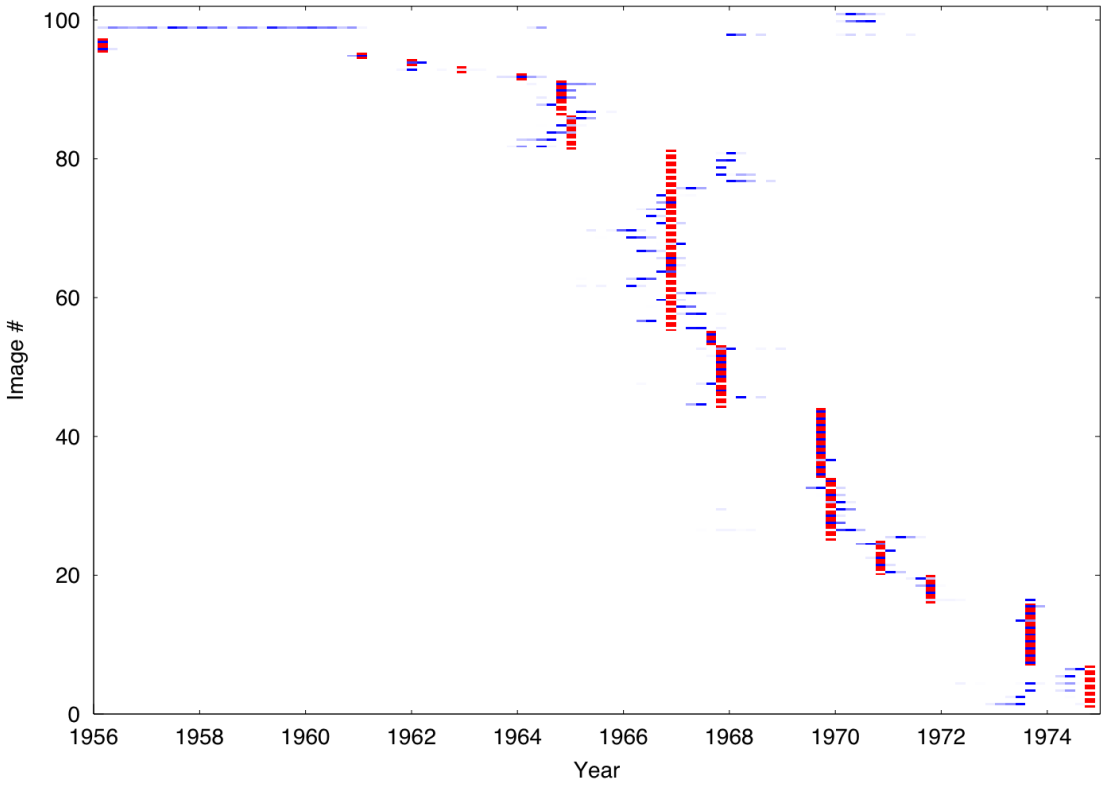


Figure 60: Marginal date distribution for each image in the Atlanta data set. Red pixels indicate initial date estimates (not ground truth, which is unavailable), while blue pixels are histograms of all MCMC samples and indicate the temporal density for each image.

pixels indicate date labels for each image (many of which are "circa" dates) since we do not have ground truth for the Atlanta data. The most surprising behavior appears at the top of the figure where the date distributions for four undated images can be seen. As described above, these represent a pair of images of the 1968 Equitable Building which have been placed in 1970, the incorrectly labeled 1868 image which was correctly moved to 1968, and an image with date estimates spread evenly between 1956 and 1961.

Providing confirmation of the validity of our probabilistic temporal inference framework is the fact that our method is able to simultaneously estimate the dates of completely undated images, adjust the dates of images with uncertain temporal information, and assign time intervals to all 3D structures for this Atlanta scene.

6.6.3 Lower Manhattan

Finally, we perform experiments on a data set consisting of 454 images of Manhattan, spanning the dates 1928 to 2010. From 83,860 points, we extract 960 buildings by computing convex hulls of point groups and extending them to an estimated ground plane. The images in this data set come from Flickr and the New York Public Library. Figures 61 through 63 depict the reconstructed 3D point cloud and the extracted 3D objects from the viewpoint of one of the images of lower Manhattan. Computing the occlusion table for Manhattan takes just under 22 minutes – though there are fewer extracted objects than in the Atlanta model, the number of images is much larger for Manhattan, leading to more rendered images during occlusion computation. We perform temporal optimization on the scene using the same experimental conditions as for Atlanta, and the estimated time intervals for each object in the scene are visualized in Figure 64.

The Manhattan data is qualitatively different from the Atlanta data set in several ways: it has more than four times as many images, there exist ground truth dates for the images, and the images span a larger period of time. Because we are mostly certain of the image dates for this dataset, the temporal inference experiment we focus on here is performing

Figure 61: Reconstructed Model of Lower Manhattan. Here we see one of the 454 images in the reconstruction. *Photo by Jimmy Hilburn.*

leave-one-out image dating to test the effectiveness of our temporal inference framework on assigning dates to real modern and historical images.

6.6.3.1 Leave-One-Out Date Estimation

We quantitatively evaluate the performance of our temporal inference framework on the Manhattan data set in a leave-one-out manner, by estimating a date for each image given known dates for all other images. In each round of this experiment, we choose one image as the test image and throw away all date information for this image. We use the given

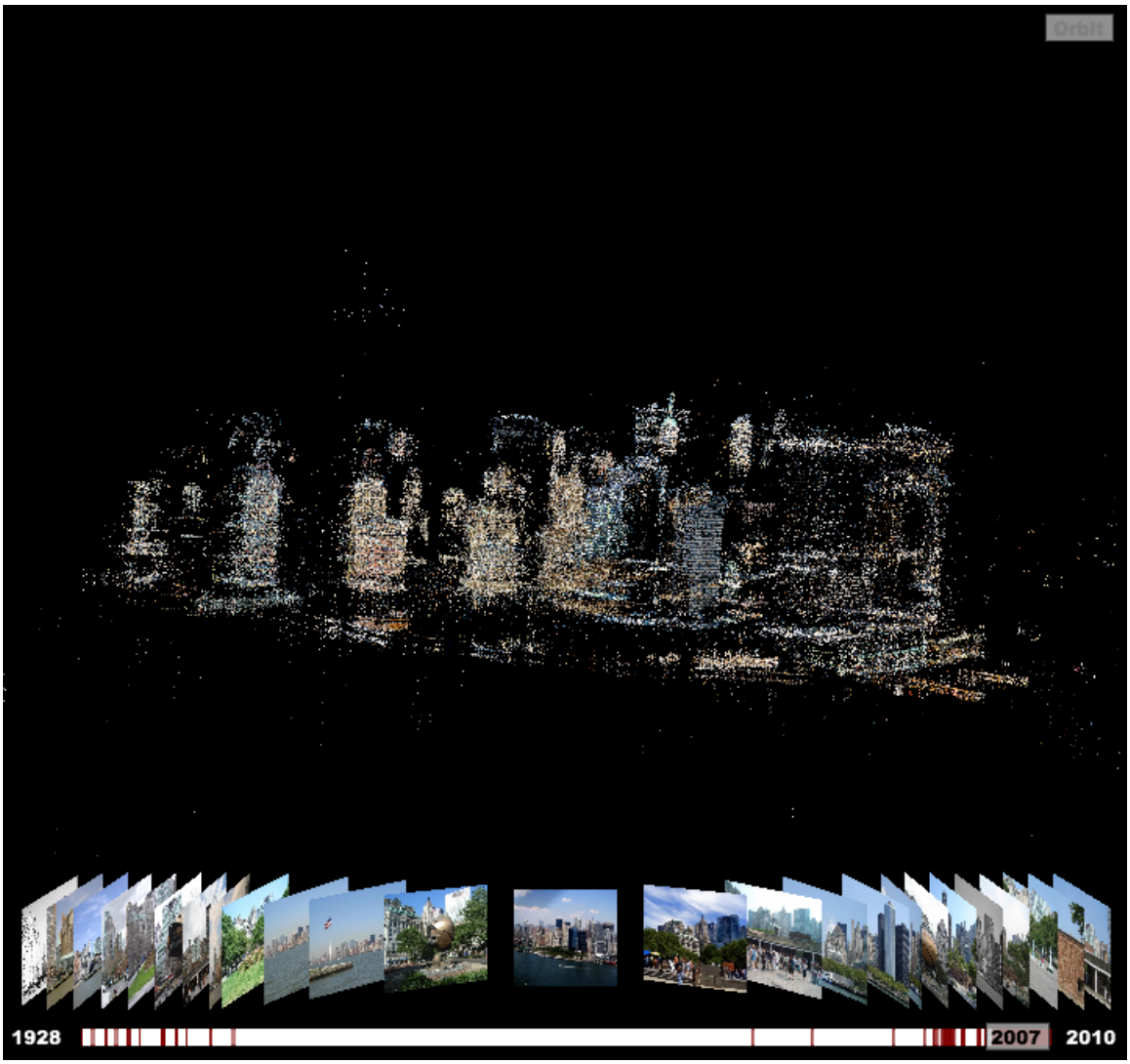


Figure 62: Reconstructed Model of Lower Manhattan. The resulting point cloud of 83,860 points from the viewpoint of the image in the previous figure.

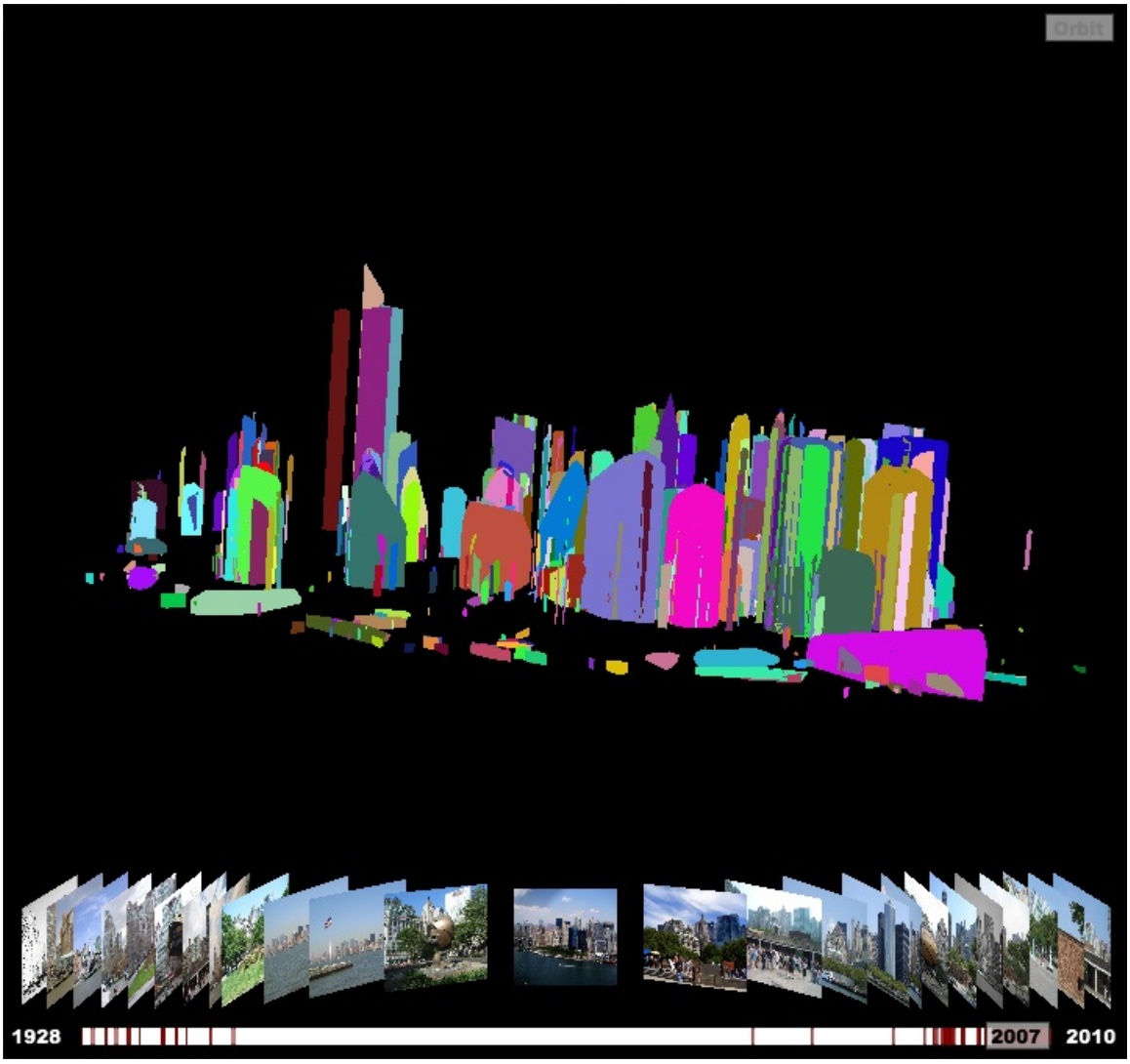


Figure 63: Reconstructed Model of Lower Manhattan. 960 objects are extracted from the point cloud in the previous image. Points are grouped according to a distance threshold and the condition of being simultaneously observed in at least one image. Convex hulls of the resulting groups are computed and extended down to an automatically estimated ground plane.

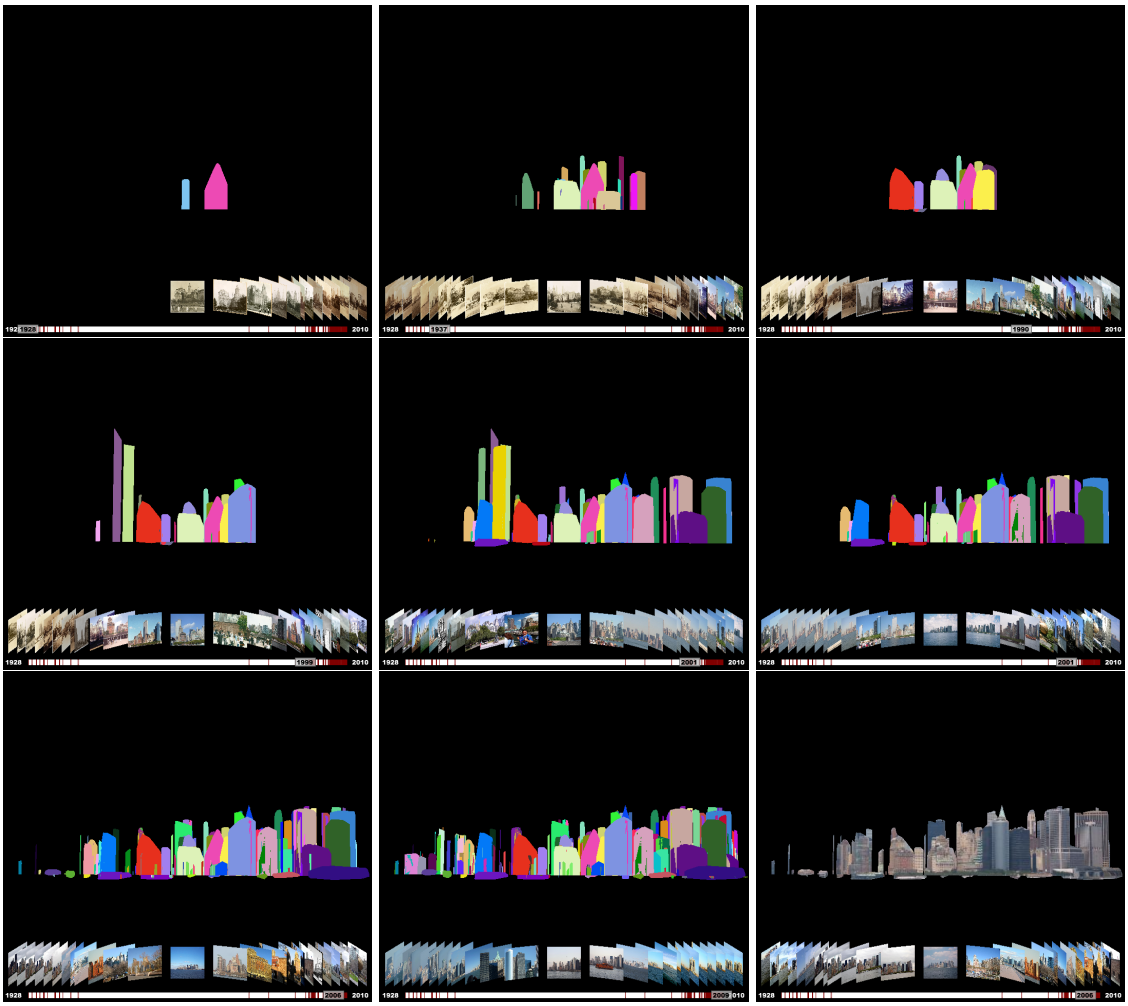


Figure 64: Manhattan 3D Geometry Over Time. Recovered time-varying 3D geometry for Manhattan. At different points in time (1928, 1937, 1990, 1999, early 2001, late 2001, 2006, and 2009), we see the automatically segmented buildings that exist at the given time. Color-coding lets us see that several of the buildings from the 1930s have survived up to the present. The bottom right figure shows an image projected onto the 3D geometry to illustrate the real-world buildings that correspond to the objects in the recovered 3D geometry.

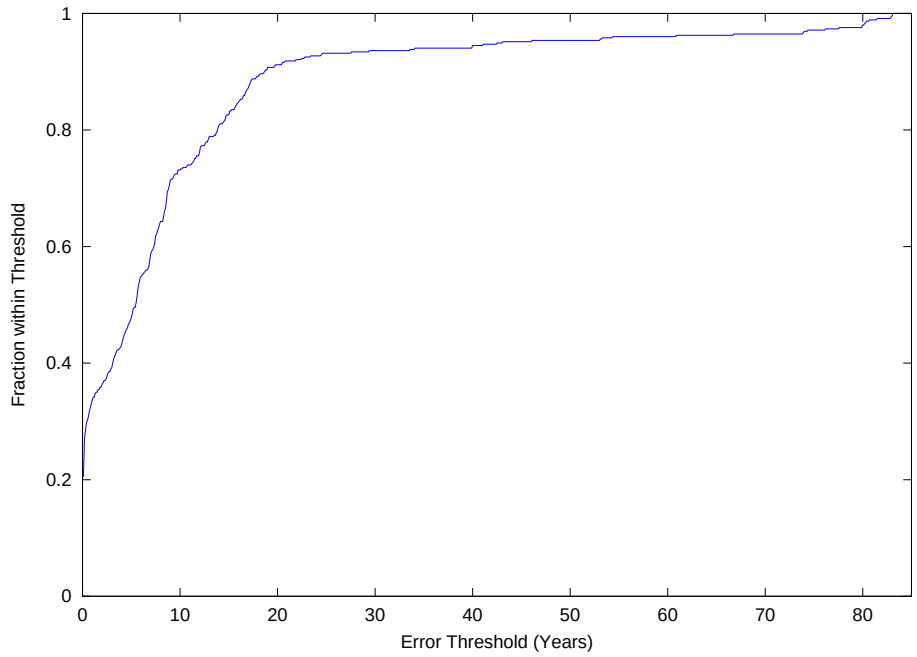


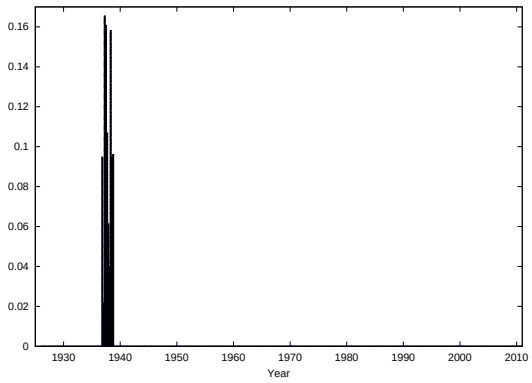
Figure 65: Summary of Date Estimation Results for Manhattan Data Set. This graph shows, for a given threshold (in years), the fraction of images with date estimates within this error threshold of their ground truth dates. Note that 34% of images are correctly dated to within a year, with 48% within 5 years, and 73% of images are dated correctly to within 10 years. This estimation is performed without using any prior date information for each test image.

dates of all other images to determine the time interval for each object, which have already been reconstructed and segmented from the resulting point cloud. In these experiments, we use the same MCMC framework as above (see Section 6.3.4.1) to perform date estimation, except that we only sample over dates for a single unknown image at a time. As a result, we get not only the maximum a posteriori date estimate, but a distribution over image dates as well.

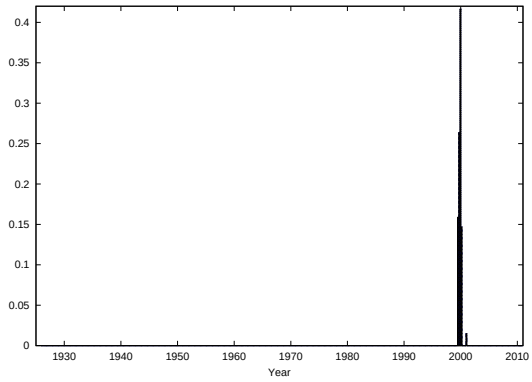
We performed leave-one-out date estimation for every image in the Manhattan data set. The results are summarized in Figure 65, which shows the fraction of images with estimated dates within a given threshold of the given date for each image. We treat these given dates as ground truth (though we know these dates only up to a given year in the case of many

older images). The result is that 34% of images are correctly dated to within a year, 48% of images are correctly dated to within 5 years, and 73% of images are dated correctly to within 10 years of ground truth. We consider this result quite good for several reasons. The fact that new buildings are not constantly being constructed means that even images taken several years apart may appear identical as far as which buildings are observed. Second, we know that not all buildings are correctly detected in all images, but these results show that our method gracefully degrades in this case. Being wrong by 10 years is a much more acceptable mistake than being wrong by 60 years. Thus, our method is making reasonable assignments based on the noisy evidence available.

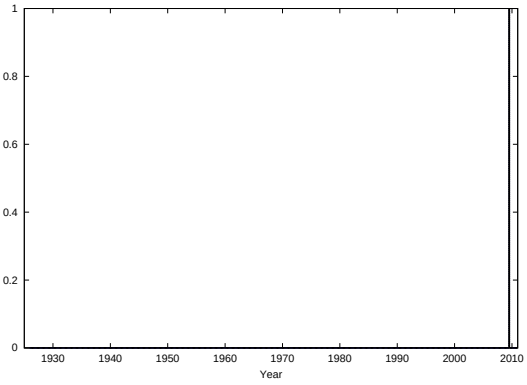
We now examine several specific instances of leave-one-out image dating in more detail. Figure 66 shows several images, along with probability distributions over the date for each image, and an estimated date and given date for each image. Note that while the probability distributions in Figure 66 have been discretized, temporal inference takes place, and MCMC samples exist, in the continuous space of image dates. In the first case we examine (Figure 66 (a)), an image from 1935 is assigned a date of 1937.2 by our temporal inference framework. In this case, we have removed all date information about the test image, such that this date assignment is purely based on maximizing the probability of the observations of objects in the image. In this case, we consider an error of 2 years a success, and in fact, the distribution over image dates for this photograph has an extent of several years surrounding the peak in 1937. In the next case we examine (Figure 66 (b)), an image of Manhattan from 2001, which includes views of the Twin Towers of the World Trade Center, is given an estimated date of 2000, which we also consider reasonable given the set of buildings present in the image. Finally, in Figure 66 (c), we see an October 2009 image which returns an estimated date of October 2009. The precision of this result can be explained by several factors. First, there is at least one building which is under construction in the image – any other images in the database which see this same building were likely



(a) Estimated Date: 1937.2 Given Date: 1935



(b) Estimated Date: 2000.5 Given Date: 2001.7



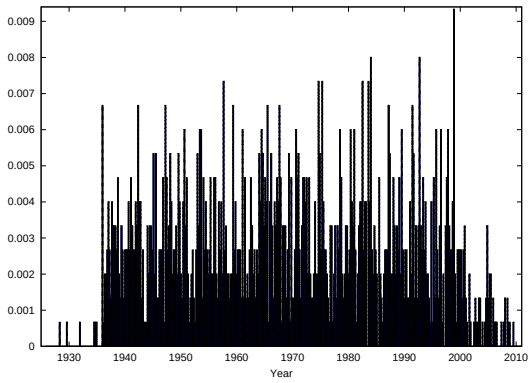
(c) Estimated Date: 2009.9 Given Date: 2009.9

Figure 66: Leave-One-Out Date Estimation. We estimate the date of a single image in the Manhattan data set given the dates of all other images. The graphs on the left display the probability that the photo on the right was taken on a given date, computed as a histogram of samples resulting from MCMC. *Photos provided by New York Public Library (a), Tony Street (b), and Kiesha Jenkins-Duffy (c).*

captured around the same time period. Related to this idea is the fact that there *are* several other images in the database captured during Fall 2009 and any temporary 3D objects may be acting as indicators of this specific point in time. As expected, this suggests that increasing the temporal density of images will lead to finer-scale accuracy in image date estimation and other temporal inference tasks.

An illustrative failure case is shown in Figure 67. The case is interesting because this particular image suffers from two problems. First, it is a highly zoomed shot showing the tops of only four buildings. Second, only one of the four buildings has been observed in the image due to failure to find SIFT matches with other images in the model for the three undetected buildings. A combination of low-contrast shadowed areas and repetitive, indistinctive features may be to blame for the detection failure. Normally, such detection failures can be overcome if there are a handful of visible buildings in the image to begin with – the buildings that are successfully detected provide reliable temporal information even when accompanied by failed detections. However, when an image with very few visible buildings has a large number of failed detections, the result, as we see in Figure 67 is that the distribution on image dates is spread out over many years. The detected building in this image is one that has existed over the entire range of dates, from the 1920s to the present, and is therefore uninformative. The reason the distribution dips in the 1930s and the 2000s is because *the model expects to see buildings in this image during those periods* and the failure to do so is evidence against the image originating from these time periods.

These leave-one-out date estimation results provide confirmation that our temporal inference framework is truly capable of recovering image dates, and therefore capable of aiding historians in a task currently carried out by hand. In other words, not only can our temporal inference framework be used to *construct* 4D city models, but once constructed these 4D city models can also be used to make *historical discoveries* about other images. Not every image is dated correctly, but now that the temporal inference model is in place, we know what can be done to improve these results. First, we must use techniques to better



(a) Estimated Date: 1998.4 Given Date: 2009.1

Figure 67: Date Estimation Failure Cases. Not all images are correctly dated due to a variety of factors, including failure to detect all buildings present in an image, and inherent ambiguity when viewing only a subset of buildings which may have existed together over a large span of time. *Photo by Flickr user kevystew*.

find correspondences between images and/or to improve detection of the presence of buildings in each image. Second, we must employ techniques to more robustly extract objects or building models from the scene. By thus improving the accuracy of the observations and the accuracy of the geometry used in visibility reasoning, we predict that the performance of our existing temporal inference method will improve as well.

Chapter VII

DISCUSSION

In this dissertation, we have demonstrated that time-varying 3D models of cities can serve to organize collections of historical and modern images, and we have introduced techniques for performing temporal inference on 3D reconstructions in order to automatically create such models. In this chapter, we begin by discussing limitations and challenges for our 4D city construction and temporal inference methods, we reiterate the contributions we have made, and we discuss possible directions for future work.

7.1 Limitations and Challenges

Despite the success of our methods for automatically constructing 4D city models, there are some limitations. When our temporal inference methods fail, the primary reason is the failure to detect and/or match all the relevant features in an image. Thus, buildings go undetected, and despite our attempts to overcome this problem with a probabilistic framework which is robust to noisy observations, the results are sometimes less than optimal (see Figure 67). This problem could potentially be addressed by a guided matching phase in which a denser set of correspondences is built upon the originally matched SIFT features. However, this would still not address the fact that a number of images fail to be included in the 3D reconstruction entirely due to a lack of feature matches between images from different time periods. We analyze this problem and consider the implications below.

7.1.1 Feature Correspondence Across Time

At the root of our methods for 4D city model creation is the notion of finding corresponding points in images taken at differing historical times (see Figure 69). Popular feature detectors and descriptors like SIFT (Lowe, 2004, 1999) are designed to be reliably detected in

differing images and invariant to changes in scale, rotation, lighting, and, to a limited extent, viewing angle. From our experience working with this problem, a primary obstacle to obtaining *unified, complete* 4D city models automatically is reliably finding features which persist across time. By *unified*, we mean a single model containing photos which span all time periods for which photographic evidence exists, and by *complete*, we mean a model in which all available images have been included in the reconstruction. If the result of structure from motion is two separate reconstructions, one built entirely from images of the 1930s and one from the 2000s, then we have failed to reconstruct a unified model. If our reconstruction is only built from 50% of the provided images, then it is not complete.

We desire a reconstructed 4D city model containing the entire photographic record a city, but using current methods, we find photographs are most likely to share corresponding points with other images taken around the same time. We have both quantitative and qualitative evidence to back up this statement.

As two images are captured farther apart in time, this introduces changes in scene structure, scene appearance, and lighting conditions, all of which may negatively impact feature matching between the two images. We also observe that, empirically, when two photographs of a scene originate from the same time period (and a roughly similar viewing direction), they share a large number of feature correspondences. We see examples of this effect in Figure 68 and note that this property is independent of the absolute date of the image, assuming similar resolution and image quality. For example two images from 1935 share 392 corresponding SIFT features, while two images taken one month apart in late 2009 and early 2010 have 251 correspondences. Taking this effect to an extreme, we see that two images taken just seconds apart in 2009 have 2367 corresponding points resulting from detecting and matching SIFT features between the two images. In the last example, this is partly a result of the very small change of viewpoint between the two images.

In practice, we do find some features that persists across time. To illustrate the effects of the passage of time on feature correspondence, see Figure 69, which shows only the very

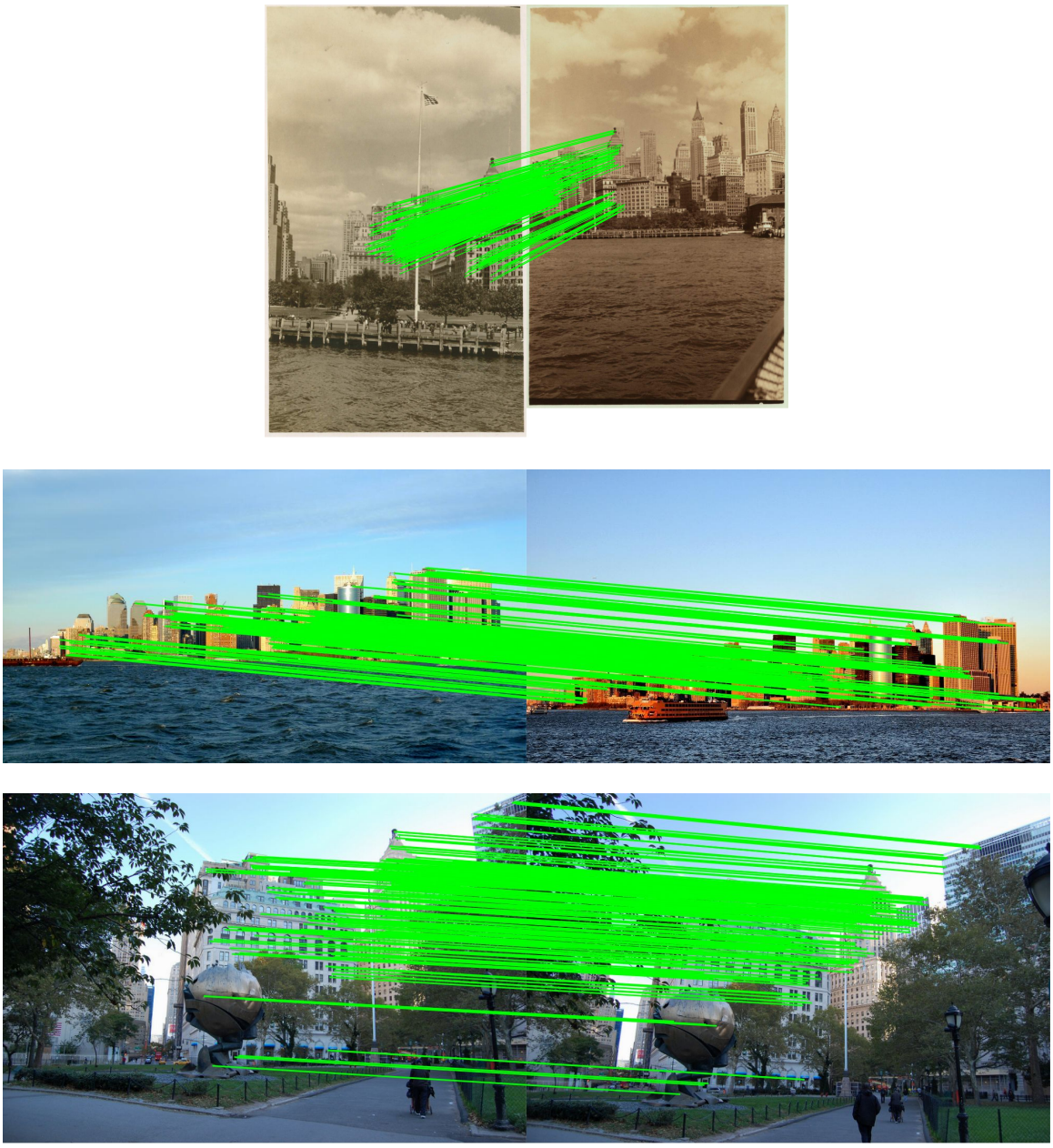


Figure 68: Matching Features. For images taken closer together in time, more geometrically consistent matching SIFT features are automatically detected. Here, we see two images taken in 1935 with 392 correspondences (top), two images taken one month apart in late 2009 and early 2010 with 251 correspondences (middle), and two images taken just seconds apart in 2009 with 2367 correspondences. *Photos provided by New York Public Library (top), Flickr user mfkne (middle left), René Alphenaar, the Netherlands (middle right), and Charles Gnilka (bottom).*

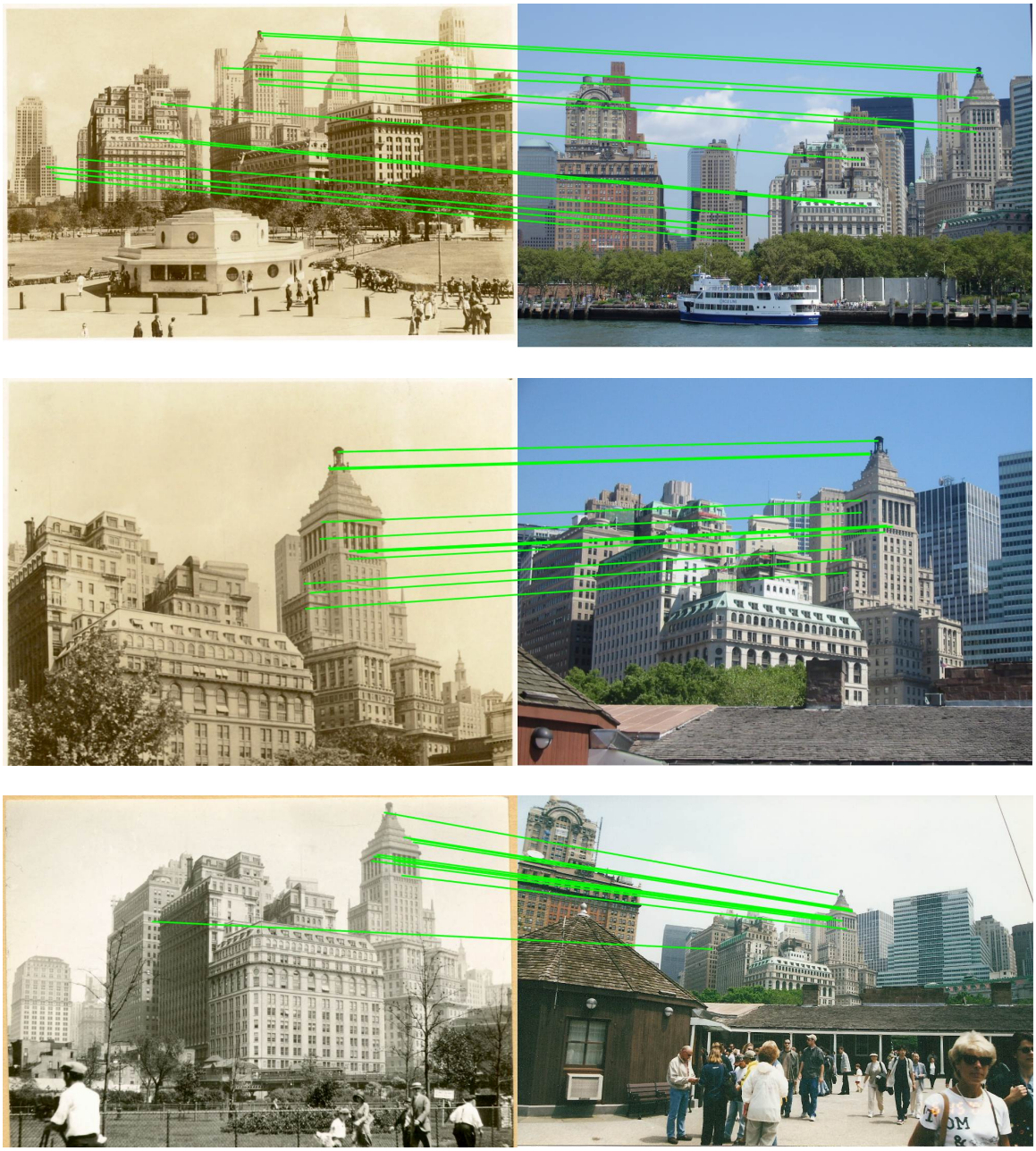


Figure 69: Matching Features Across Time. Relatively few feature matches are found in images of the same location, but separated by decades of time. Successfully matched images in such cases usually involve extremely similar viewpoints and lighting conditions, and we still achieve around 20 matches at best. Here we see an image pair from 1936 and 2007 with 17 correspondences (top), a pair from 1936 and 2009 with 19 correspondences (middle), and a pair from 1929 and 2000 with 16 correspondences (bottom). *Photos provided by New York Public Library (left), Ray Kippig (top right), Tony Street (middle right), and Robert Schoneman (bottom right).*

best sets of matches between images captured around 70 years apart. These image pairs contain only 16, 17, and 19 point correspondences as identified by matching SIFT features, while a majority of other images we tested from the 1920s and 1930s do not contain any significant number of geometrically consistent matches to more modern images. (During a fundamental matrix-based RANSAC stage to check for geometrically consistent matches, we use a threshold of 16 inliers to eliminate spurious matches.)

To quantify this effect across a database of 454 images of Manhattan, we detect SIFT features in each image, find geometrically consistent matches in all other images, and for every remaining point correspondence, we determine the number of years across which this match occurred. We then create a histogram showing how many feature matches occurred for each given value of time difference. In Figure 70, we show these plots for both Atlanta (102 images over roughly 20 years) and Manhattan (454 images over roughly 80 years). There is a clear inverse relationship between the number of years separating two images and the number of feature correspondences found between them. Note that if it was strictly true that the probability of finding feature correspondences between two images (of the same scene) is directly related to the amount of time which has passed between them, then we could derive an estimate of how densely sampled we require our photographs to be, in time, in order to construct a unified model of the scene. We leave this analysis to future work.

The decreased effectiveness of feature-matching with the passage of time presents us with a problem if we want to acquire unified, complete 4D city models automatically. One fundamental question is this: Do we need to design new features that are time-invariant or do we simply need to collect enough data that current methods suffice? For example, the pattern of repeating windows on a building (Schindler et al., 2008) might be one possible time-invariant feature. However, in this work, we chose to adopt SIFT features, leaving the exploration of time-invariant features to future work. So we must examine this problem from the point of view of the density of available data. If we had dense photographic

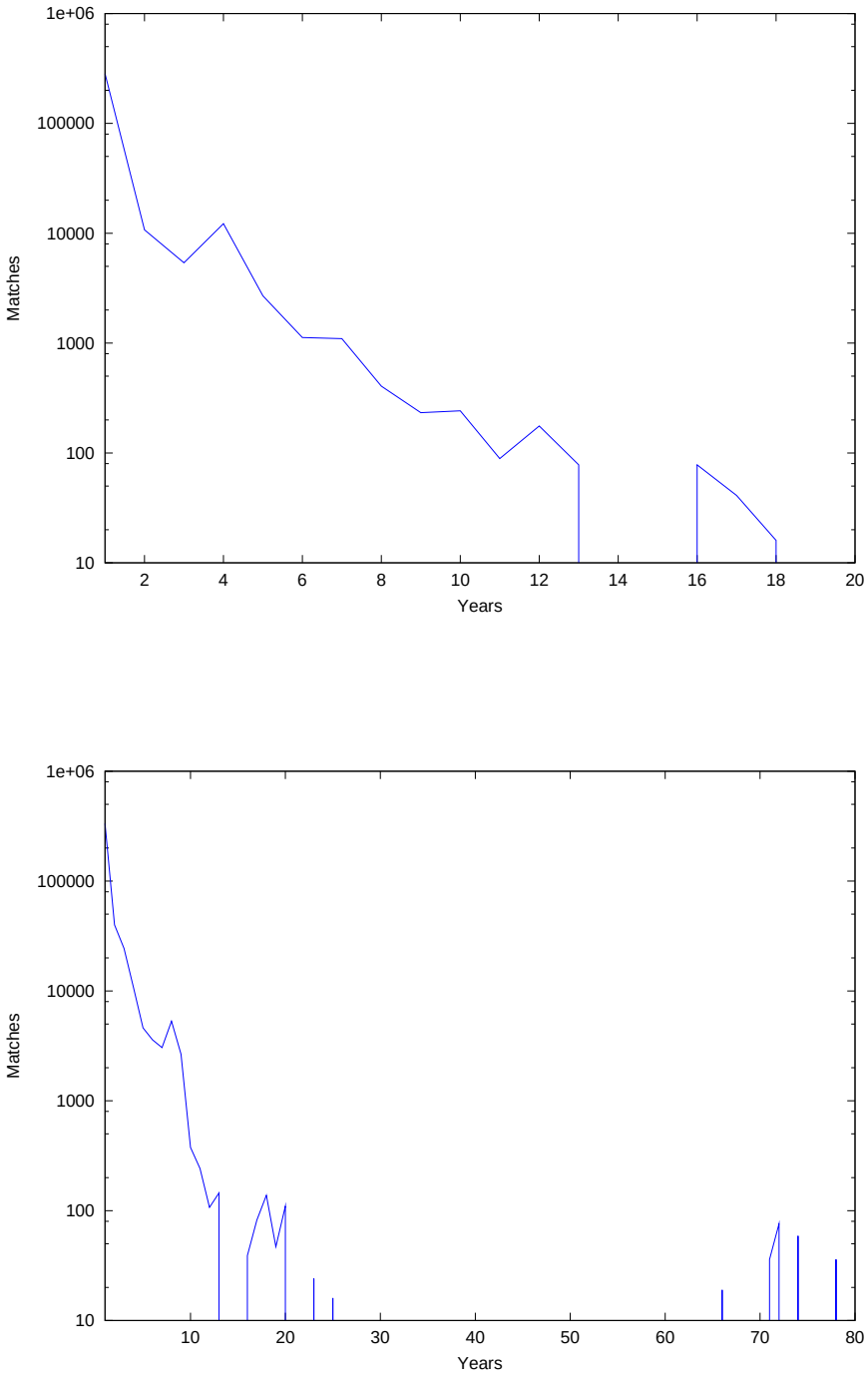


Figure 70: Plot of Feature Matches Across Time. We plot, on a log scale, the number of geometrically consistent feature matches against the time difference between the two images in which each feature match occurs. On top, the plot for Atlanta, and on bottom, for Manhattan. Gaps in the plot are partially due to the scarcity of images with the corresponding time separation, but are also due to lack of matches even when such image pairs exist.

coverage of a city in both space and time, would feature correspondence still be a problem? If it is true that two photographs taken one day apart from roughly the same viewpoint will have numerous feature correspondences with high probability, then we could indeed solve this problem by using densely sampled data. Unfortunately, the photographic record (at least the portion available to us) is not complete and has not been uniformly sampled in time. This leads to several fundamental questions: What is the temporal density of photographs that exist, in some form, for a given city? How many of these photographs are actually available? In the case of both Atlanta and Manhattan, we have performed our experiments on the order of several hundred historical images that we have been able to acquire for each city. While sufficient for the purposes of this dissertation, the success of 4D city models as a means of organizing the world's photographs will depend upon gaining access to a much larger collection of historical images.

7.1.2 Uniting Modern and Historical Reconstructions

To illustrate the difficulties inherent in uniting modern and historical images into a unified 4D model, we examine the pattern of matches between a set of photos of Atlanta, some of which were captured in 2008 and some of which were captured in the 1950s, 1960s, and 1970s. Figure 71 depicts the match table describing the number of matching features between every pair of images in the data set. Darker red squares indicate more matches, while white squares indicate fewer than 16 geometrically consistent matches (we adopt a threshold to filter out false positive consistent sets of matches, as in (Snavely et al., 2006)). The two triangular sets of matching images are divided cleanly between the set of modern images and historical images, with no image pairs achieving a geometrically consistent set of matches greater than the threshold of 16. We also found that lowering this threshold does begin to admit spurious matches into the solution without uniting the modern and historical images properly.

Thus, we end up with two separate 3D reconstructions of Atlanta, despite the fact that

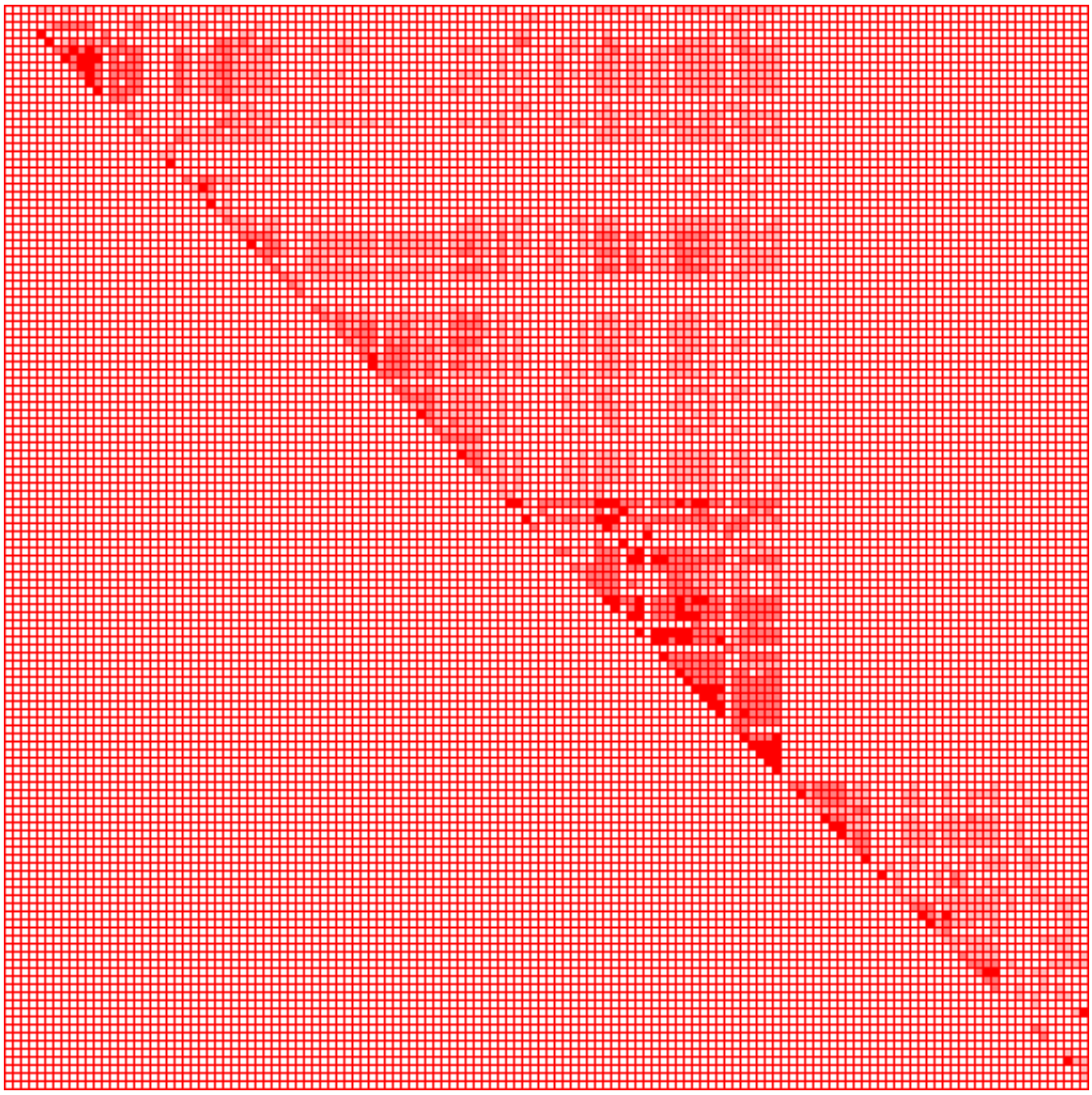


Figure 71: Match Table for Modern and Historical Images of Atlanta. This match table describes the number of matching features between every pair of images in the data set. Dark red squares indicate greater than 1000 matches between the two images represented by a specific row and column of the table. Medium red squares indicate greater than 100 matches, and light red squares indicate greater than 16 matches. White squares indicate fewer than 16 geometrically consistent matches, which we is the threshold we adopt in this work. All matches are counted after a RANSAC step to determine geometric consistency. The two triangular structures in the table correspond to the matches between modern images (left side) and matches between historical images (right side), with no matches linking the two components together.

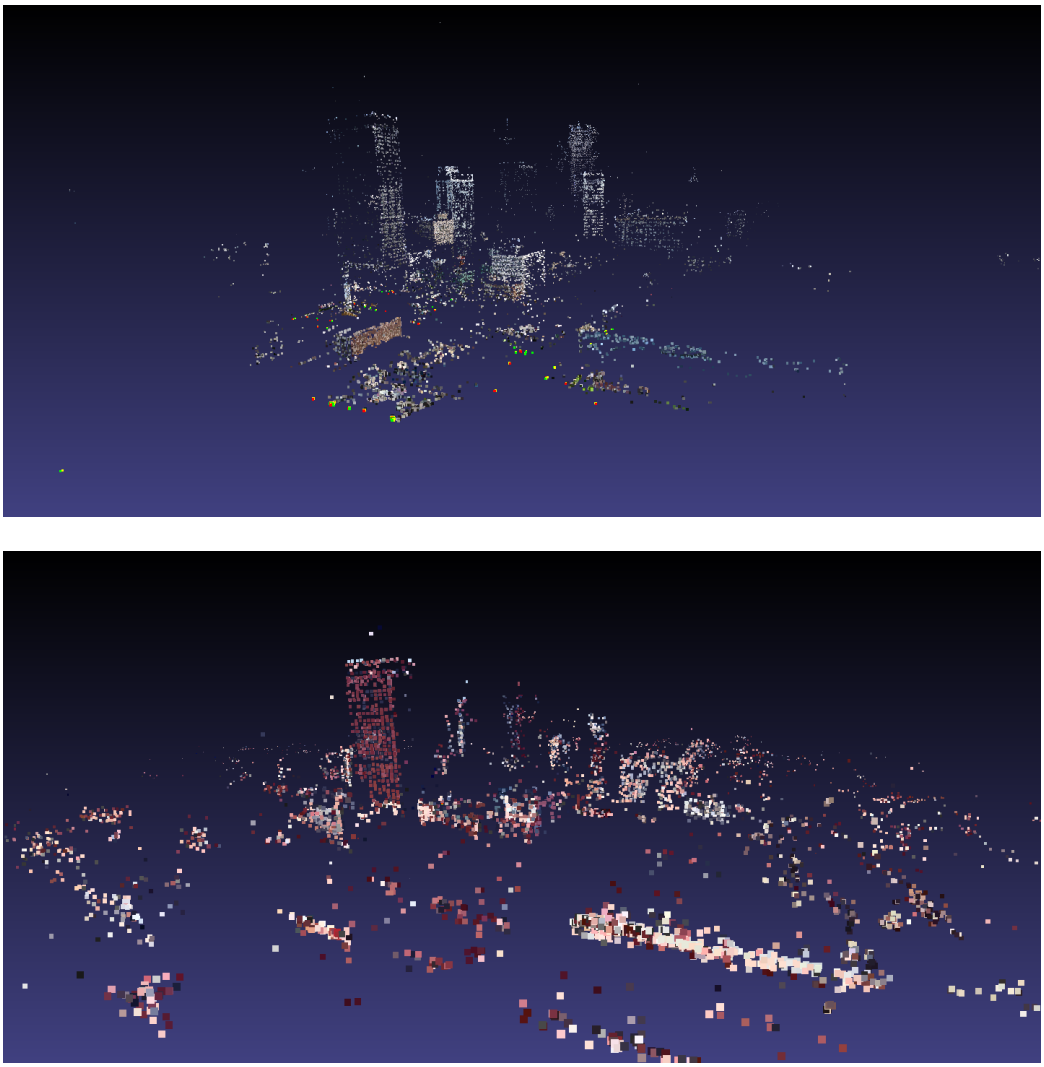


Figure 72: Modern and Historical Reconstructions of Atlanta. Because no images in the data set provide matches linking old and new images, we end up with two separate 3D reconstructions (2008 reconstructed point cloud on top, 1950s-1970s reconstruction on bottom), and we are unable to create a united 4D model of the city, despite the fact that both reconstructions depict overlapping sets of buildings. One solution to this problem is to collect more data, a difficult task in the case of historical imagery.

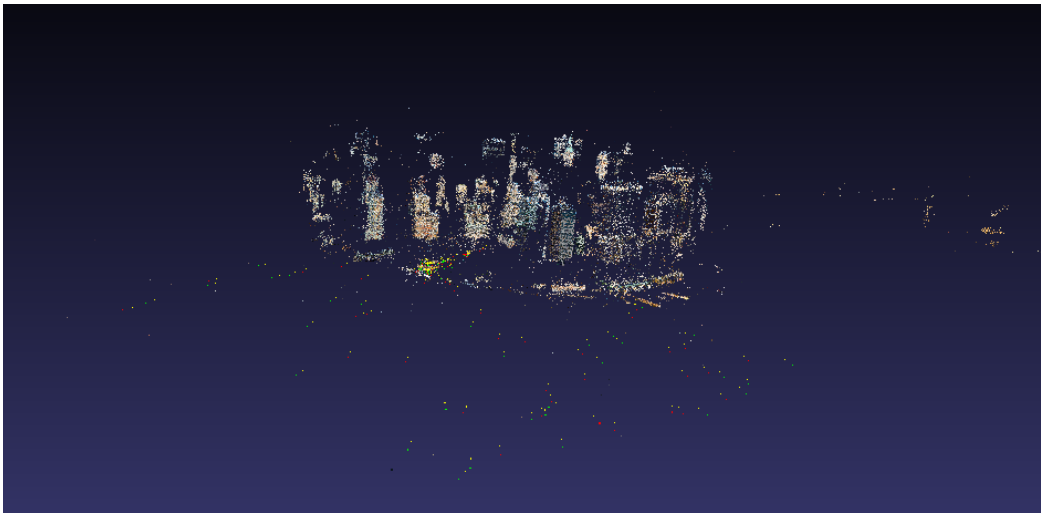


Figure 73: United Modern and Historical Reconstruction of Lower Manhattan. Because SIFT feature matches were found in common between modern and historical images of lower Manhattan, we are able to build a united 3D reconstruction which we use as the basis for a 4D model of the city.

both reconstructions depict some of the same buildings (see Figure 72). All attempts to acquire images that would match to both the modern and historical imagery for Atlanta (and thus unite the two reconstructions) met with failure. The Atlanta data set we focused on in Chapter 6 consists of the 1950s, 1960s, and 1970s Atlanta images only. On the other hand, the Manhattan dataset shown in Figures 68 and 69 was successfully united into a single reconstructed 3D model (Figure 73) due to the existence of a small number of matches between modern and historical imagery.

Thus, given our experience, we see the difficulty of matching features across time as one of the primary limiting factors of our presented methods, and as one of the major obstacles to automatic construction of *unified* and *complete* 4D city models.

7.2 Contributions

At the beginning of this dissertation, we stated our thesis in two parts. The first part of the statement deals with the motivation behind building 4D city models in the first place: *4D city models serve to both organize and enhance the world's historical photographs by providing spatial and temporal context for every image.*

We supported this claim in Chapters 2 and 3 by demonstrating how we have been able to use 4D city models of Atlanta, Seoul, and Manhattan to organize and explore historical photo collections and to make historical discoveries. The time-based methods of interacting with 4D models which we have introduced in this dissertation represent a significant advancement in the way we experience historical photo collections.

The second part of the thesis statement is as follows: *Temporal inference algorithms, when applied to reconstructed 3D scenes, enable the automatic construction of 4D city models directly from images.*

We demonstrated the truth of this statement in Chapters 4, 5, and 6 by defining the temporal inference problem and presenting three methods for solving it. First, we showed that reasoning about the visibility of objects in a scene enables the recovery of a temporal ordering of images of the scene. We then introduced mechanisms for incorporating dates into the temporal inference process, dealing with the uncertainty inherent in historical date labels. Next, we introduced a probabilistic formulation of the temporal inference problem that combines visibility reasoning with uncertain image dates to arrive at an optimal solution for all time parameters, including a date for each image and a time interval for each object in the scene. Finally, based around these temporal inference methods, we demonstrated a fully automated pipeline for building 4D city models from images.

To reiterate, the fundamental contributions of this dissertation have been:

- developing a formal representation of time in structure from motion problems
- presenting three algorithms for solving temporal inference
- detailing a pipeline for automatically building 4D city models
- introducing a method of interacting with 4D models
- demonstrating 4D models of Atlanta, Manhattan, and Seoul.

The experiments we perform with our Manhattan and Atlanta data sets ultimately show the power of our model construction and probabilistic temporal inference methods. From a set of input images, we are able to build time-varying 3D models automatically, to find a temporal solution for all image dates and building time intervals, and to assign dates to undated images. Finally, we are able to actually use these automatically constructed models for 4D city interaction, enabling exploration of our historical photo collections and providing spatial and temporal context to all the images.

7.3 Future Work

The natural endpoint of this line of research is to register every photograph in existence, across all time periods, to a common global reference frame and to use this exhaustive photographic record to construct a time-varying 3D model of the world that is as accurate and complete as can be achieved from photographic evidence. Such a comprehensive model would serve as a general reference tool for the visual world, much as Wikipedia or Google Earth are used today. Such a model would not only allow a person to find historical images similar to their own modern photographs, but with further research, to shoot a video and see what it would look like in a different time period, and even to walk around in the present, using a mobile phone as a window into the past via real-time visualization of one's current viewpoint from any time in history.

There are a number of important problems to be solved that would benefit any attempts to reach this ambitious goal. At the low level, designing features that are time-invariant could greatly improve the ability of images to be incorporated into a 4D model in the first place. Higher up the chain, if we have methods of obtaining more accurate building models from segmented point clouds, we would improve both the accuracy of visibility reasoning and the quality of resulting 4D city visualizations. At the visualization level, a future goal is to be able to really dive into a single image of a city at any point in history, combining the appearance of the parts of the scene visible in the given image and the known 3D geometry

of the rest of the scene to create a convincing reconstruction of the world at a moment in time.

Finally, one of the major obstacles to reaching these goals is simply getting access to the historical imagery necessary for constructing 4D city models. It is our hope that, as time goes on, more and more of these historical images will become freely available online and that the methods described in this dissertation will be used to truly unlock the urban photographic record of our world.

Bibliography

S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Intl. Conf. on Computer Vision (ICCV)*, 2009.

J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11): 832–843, 1983.

S. Badaloni, M. Falda, and M. Giacomin. Integrating quantitative and qualitative fuzzy temporal constraints. *AI Commun.*, 17(4):187–200, 2004.

J. Bauer, K. Karner, K. Schindler, A. Klaus, and C. Zach. Segmentation of building models from dense 3d point-clouds. In *Proc. 27th Workshop of the Austrian Association for Pattern Recognition*, pages 253–258. Citeseer, 2003.

R. Cipolla, D. Robertson, and E. Boyer. Photobuilder - 3D models of architectural scenes from uncalibrated images. In *ICMCS, Vol. 1*, pages 25–31, 1999.

P. Debevec, Y. Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering*, pages 105–116, 1998.

P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *SIGGRAPH*, 30:11–20, 1996.

R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49 (3):61–95, May 1991.

A.R. Dick, P.H.S. Torr, and R. Cipolla. A Bayesian estimation of building shape using MCMC. In *Eur. Conf. on Computer Vision (ECCV)*, pages 852–866, 2002.

D. Dubois, H. Fargier, and H. Prade. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence*, 6:287–309, 1996.

D. Dubois, H. Fargier, and P. Fortemps. Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research*, 147: 231–252, 2003.

O. D. Faugeras, E. Le Bras-Mehlman, and J. D. Boissonnat. Representing stereo data with the Delaunay triangulation. *Artif. Intell.*, 44(1-2):41–87, 1990.

O.D. Faugeras and Q.T. Luong. *The geometry of multiple images*. The MIT press, Cambridge, MA, 2001. with contributions from T. Papadopoulo.

Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Intl. Conf. on Computer Vision (ICCV)*, 2009.

M. Ge and M. D'Zmura. 4D structure from motion: a computational algorithm. In *Computational Imaging.*, pages 13–23, June 2003.

M. Golparvar-Fard, F. Peña-Mora, and S. Savarese. Monitoring of construction performance using daily progress photograph logs and 4d as-planned models. In *ASCE International Workshop on Computing in Civil Engineering*, 2009.

A. Griewank. On Automatic Differentiation. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developments and Applications*, pages 83–108. Kluwer Academic Publishers, 1989.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

D. Jelinek and C. J. Taylor. View synthesis with occlusion reasoning using quasi-sparse feature correspondences. In *Eur. Conf. on Computer Vision (ECCV)*, pages 463–478, 2002.

S.B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *Intl. J. of Computer Vision*, 38(3):199–218, 2000.

C. Schmid L. Lazebnik and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004.

D.G. Lowe. Object recognition from local scale-invariant features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1150–1157, 1999.

P. Messier. Notes on dating photographic paper. *Topics in Photograph Preservation*, 11, 2005.

Halvor Moorshead. *Dating Old Photographs 1840-1929*. Moorshead Magazines Ltd, 2000.

D. D. Morris and T. Kanade. Image-consistent surface triangulation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 332–338, 2000.

P. Mueller, G. Zeng, P. Wonka, and Luc Van Gool. Image-based procedural modeling of building facades. In *SIGGRAPH*, volume 26, New York, NY, USA, 2007. ACM Press.

Nando de Freitas Pinar Duygulu, Kobus Barnard and David Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, pages 97–112, 2002.

M. Pollefeys, D. Nistér, J.M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2):143–167, 2008.

Robert Pols. *Family Photographs, 1860-1945: A Guide to Researching, Dating and Contextuallising Family Photographs*. Public Record Office Publications, 2002.

Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

Laura Walker Renninger and Jitendra Malik. When is scene recognition just texture recognition? *Vision Research*, 44:2301–2311, 2004.

G. Schindler and F. Dellaert. Line-based structure from motion for urban environments. In *3D Data Processing Visualization and Transmission (3DPVT)*, 2006.

G. Schindler, F. Dellaert, and S.B. Kang. Inferring temporal order of images from 3D structure. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

S.M. Seitz and C.R. Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30. ACM, 1996.

L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2048, 2006.

N. Snavely. Bundler: Structure from motion for unordered image collections. http://phototour.cs.washington.edu/bundler/, 2008.

N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

R. Sosic and J. Gu. 3,000,000 queens in less than one minute. *SIGART Bull.*, 2(2):22–24, 1991.

C. J. Taylor. Surface reconstruction from feature based stereo. In *Intl. Conf. on Computer Vision (ICCV)*, page 184, 2003.

A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. In *IEEE Trans. Pattern Anal. Machine Intell.*, volume 30, pages 1958–1970, 2008.

B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, Sep 1999.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

G. Wolberg. Image morphing: a survey. *The Visual Computer*, 14(8):360–372, 1998.

L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Eur. Conf. on Computer Vision (ECCV)*, 2008.